



Conversor de texto a pictogramas

Proyecto de Sistemas Informáticos

Curso 2013/2014

Directores: Raquel Hervás Ballesteros y Susana Bautista Blasco

Autores: Agustín Hernández-Gil Alonso, Carlos Martínez Díaz y Sergio Pascua García

Departamento de Ingeniería del Software e Inteligencia Artificial

Facultad de Informática

Universidad Complutense de Madrid



Autorización de difusión y utilización

Los autores del presente proyecto Agustín Hernández-Gil Alonso, Carlos Martínez Díaz y Sergio Pascua García autorizamos a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, los contenidos audiovisuales incluso, si incluyen imágenes de los autores, la documentación y/o el prototipo desarrollado.

Asimismo todos los pictogramas utilizados en el sistema desarrollado forman parte de la colección ARASAAC, son propiedad del Gobierno de Aragón y han sido creados por Sergio Palao, se publican bajo Licencia Creative Commons (BY-NC-SA), autorizándose su uso para fines sin ánimo de lucro siempre que se cite la fuente, autor y se compartan bajo la misma licencia.

**Fdo. Agustín Hernández-Gil
Alonso**

**Fdo. Carlos Martínez
Díaz**

**Fdo. Sergio Pascua
García**



Dedicatoria

Este proyecto está dedicado a nuestras familias y a todas aquellas personas que han estado a nuestro lado y han hecho posible la realización del mismo, ya sea ayudándonos, apoyándonos o simplemente animándonos en los momentos difíciles.



Agradecimientos

Queremos agradecer a ARASAAC (Portal Aragonés de la Comunicación Aumentativa y Alternativa) su colaboración al habernos proporcionado material pictográfico sin el cual habría sido imposible la realización del presente proyecto. Así como su disponibilidad a la hora de atender consultas y preguntas.

A Lorenzo González, profesor de formación profesional en el centro de MISLATA, y a Miguel Martínez por su colaboración en el cambio de arquitectura realizado en el proyecto.

A Paula Pascua por su ayuda con los textos en inglés.

A Laura Fernández por su colaboración a la hora de realizar las pruebas.

Y por último pero no menos importante a Raquel Hervás y Susana Bautista, por ayudarnos, guiarnos y facilitarnos la realización de la totalidad del proyecto ya que sin ellas hubiese sido mucho más difícil.



Resumen

En la actualidad existe una gran cantidad de personas que presentan problemas a la hora de comunicarse mediante texto escrito. Este problema supone un alto riesgo de exclusión social para este grupo de personas al no tener acceso a una forma de comunicación clave como es la escritura. Además esta forma de comunicación ha cobrado mayor relevancia en relación con las nuevas tecnologías.

El objetivo del proyecto se centra por tanto en proporcionar una herramienta que facilite la comunicación de personas que no presentan dificultad en la comunicación escrita con personas que sí lo hacen, realizando la traducción de texto convencional a pictogramas.

Para ello hemos realizado un servicio web que permite su utilización independientemente de la tecnología del cliente que lo solicite, de forma que tenga mayor alcance dentro de la sociedad. Asimismo hemos realizado un cliente Android, escogido por ser una de las tecnologías más extendidas en este ámbito, que se sirve de la funcionalidad de dicho servicio web para llevar a cabo la traducción.

Del mismo modo, hemos llevado a cabo una evaluación de nuestro sistema que corrobora que efectivamente la traducción que proporciona es correcta y altamente adaptada a las necesidades de las personas a las que va dirigida.

Finalmente hemos querido que este proyecto sea de software libre conformando así una herramienta altamente accesible para la sociedad.



Abstract

Nowadays, there is a significant amount of the population that have problems when communicating through written text. This means that many people are in high risk of social exclusion as they find themselves without access to a key form of communication such as writing. Besides, this form of communication has become more relevant in relation to new technologies.

This project, focuses on providing a tool that facilitates communication between people who have difficulty in writing or understanding written text, and people who do not face any issues. Our solution is a tool that translates conventional text to pictograms.

We have made a web service that can be used by anyone regardless of each individual, software or device of choice, so that it has greater reach in society. Bearing better access for everybody in mind, Android technology was chosen for being one of the most wide-spread technologies in this field to make our client/app, which uses the functionality of the web service to perform the translation.

An evaluation of our system has been carried out, which effectively confirms that the translation is not only accurate but also provides a service tailored to the needs of the people to whom it is addressed.

Finally, we wanted to make this project open source thus forming an accessible tool society could benefit from.



Palabras clave

- Traductor
- Pictogramas
- Servicio web
- REST
- Android
- Spring
- Hibernate
- AraTraductor



Keywords

- Translator
- Pictograms
- Web Service
- REST
- Android
- Spring
- Hibernate
- AraTraductor



Índice

| | |
|--|-----------|
| Autorización de difusión | 2 |
| Dedicatoria | 3 |
| Agradecimientos..... | 4 |
| Resumen | 5 |
| Abstract..... | 6 |
| Palabras clave | 7 |
| Keywords..... | 8 |
| Índice..... | 9 |
| 1. Introducción | 12 |
| 1.1 Motivación..... | 12 |
| 1.2 Objetivos | 12 |
| 1.3 Gestión del proyecto | 13 |
| 2. Estado del arte | 18 |
| 2.1 Pictogramas | 18 |
| 2.2 SAAC | 18 |
| 2.3 Aplicaciones existentes basadas en pictogramas | 21 |
| 2.4 Tecnología de servicios web..... | 26 |
| 3. Herramientas utilizadas | 34 |
| 3.1 Android..... | 34 |
| 3.2 Base de Datos ARASAAC..... | 34 |
| 3.3 Maltparser | 36 |
| 4. Arquitectura del sistema | 39 |
| 4.1 Descripción general de la arquitectura | 39 |
| 4.2 Implementación de la arquitectura | 40 |
| 5. Base de datos..... | 44 |
| 5.1 Descripción de las tablas | 44 |
| 5.2 Creación de la base de datos, en formato SQL | 45 |
| 6. Traductor de texto a pictogramas | 48 |
| 6.1 Funcionalidad | 48 |



| | |
|--|------------|
| 6.2 Implementación | 49 |
| 6.3 Implementación del traductor como servicio web | 51 |
| 7. AraTraductor | 56 |
| 7.1 Funcionalidades implementadas | 56 |
| 7.2 Detalles de implementación | 61 |
| 8. Evaluación | 64 |
| 8.1 Gold standard | 64 |
| 8.2 Evaluación de los resultados | 65 |
| 8.3 Discusión de los resultados obtenidos | 75 |
| 8.4 Indicadores y métricas de los resultados | 77 |
| 9. Conclusiones y trabajo futuro | 84 |
| 9.1 Discusión | 84 |
| 9.2 Conclusiones | 85 |
| 9.3 Conclusiones y trabajo futuro | 85 |
| 10. Apéndices | 89 |
| 10.1 Apéndice A: Pruebas realizadas | 89 |
| 11. Bibliografía | 121 |



Capítulo 1: Introducción

1. Introducción

1.1. Motivación

Actualmente, el ámbito tecnológico tiene cada vez un peso mayor en la participación e intervención social. Por este motivo, muchas personas ven limitado su derecho fundamental a esta participación, ya que estas nuevas tecnologías presentan grandes barreras para personas con dificultades en la comunicación.

Estas barreras giran fundamentalmente entorno a los siguientes puntos:

- Su diseño no tiene en cuenta las dificultades de este tipo de usuarios.
- Cada producto utiliza su propio sistema de comunicación sin ningún estándar que se pueda seguir.
- Los sistemas creados presentan incompatibilidades.
- Ausencia de herramientas de traducción a otros sistemas de comunicación que faciliten su uso a los usuarios.

De este modo, esta situación supone un grave riesgo de exclusión social para estas personas que presentan dificultad en la comunicación, por lo que el presente proyecto pretende tender un puente entre aquellas personas que no presentan ningún tipo de dificultad en la comunicación, pero desean hacerse entender con personas que si lo tienen.

Por estos motivos este proyecto nace con la idea de crear una herramienta que permita a las personas no habituadas al uso de sistemas de comunicación alternativos establecer una comunicación clara para ambos interlocutores.

1.2. Objetivos

El presente proyecto tiene como meta la realización de una herramienta completa, accesible y de fácil uso, que permita realizar la traducción de texto a pictogramas. Este objetivo está motivado por la consecución de una importante mejora en la comunicación entre personas que no presentan problemas a la hora de leer y comprender texto escrito pero que no están familiarizados con el uso de

pictogramas, y aquellas a las cuales les resulta de gran ayuda la utilización de dichos pictogramas.

Para conseguir la herramienta deseada se han fijado los siguientes objetivos:

- La herramienta debe proporcionar una traducción unívoca y sin ambigüedad.
- La herramienta debe ser completamente accesible, por lo que la aplicación ha sido desarrollada para la plataforma Android presente en móviles, *tablets* y otros dispositivos similares.
- Del mismo modo, la aplicación utiliza los pictogramas de ARASAAC que componen un estándar muy extendido, siendo de este modo más sencillo que pueda llegar a un número de usuarios mayor.
- La herramienta se ha desarrollado de forma que sea de uso sencillo proporcionando incluso la posibilidad de que se introduzca el texto a traducir no sólo escribiéndolo sino también por voz.
- La aplicación permite que sus utilidades sean compartidas con otras herramientas y aplicaciones Android.

De este modo, cumpliendo con los objetivos expuestos, el proyecto proporciona una herramienta que supone un amplio avance en la comunicación entre personas sin conocimiento en el uso de pictogramas y personas a las que estos pictogramas les resultan de gran ayuda para comprender texto escrito.

1.3. Gestión del proyecto

Para organizar y gestionar la ejecución del proyecto nos hemos apoyado en varias herramientas. Para la implementación del mismo hemos utilizado el lenguaje de programación Java y en concreto el compilador Eclipse. Asimismo hemos utilizado Google Code para almacenar y compartir nuestro código entre nosotros.

Por otro lado y de forma paralela toda la documentación generada en el proyecto se ha almacenado en Google Drive, plataforma de almacenamiento en la nube.

Por último es importante destacar que la ejecución de este proyecto ha sufrido tres iteraciones que han evolucionado a su vez según aprendíamos,

conocíamos y definíamos el mismo proyecto en sí. Es por ello que a continuación se pasa a comentar en detalle estas tres iteraciones:

1.3.1. Iteración 1

Objetivos:

- ✓ Adquirir conocimiento sobre los Sistemas Aumentativos y Alternativos de Comunicación (SAAC), así como de tecnologías de servicios web y aplicaciones Android.
- ✓ Comienzo de la memoria.
- ✓ Creación de la base de datos de pictogramas.
- ✓ Obtención de un primer prototipo de la aplicación.

Desarrollo:

En la primera iteración entramos en contacto con el mundo de los SAAC, más en concreto con el de los pictogramas, aprendiendo la finalidad y utilidad de estos, así como su forma de uso. Asimismo tuvimos que aprender a desarrollar servicios web, para lo que comenzamos utilizando Axis2, un motor para servicios web y, del mismo modo, aprender a desarrollar aplicaciones para el sistema operativo de dispositivos móviles, Android.

Al mismo tiempo se creó la primera versión de la memoria del proyecto en la que básicamente volcábamos los conocimientos que íbamos adquiriendo y que pensamos que serían parte importante durante el desarrollo del proyecto.

Por otro lado, tras ponernos en contacto con la asociación ARASAAC (ARASAAC s.f.), nos proporcionaron una base de datos en Excel, que contenía palabras y sus pictogramas asociados. Partiendo de esta base se desarrolló, como se comenta más en detalle en el capítulo 6, una base de datos propia SQL adaptada perfectamente a nuestras necesidades.

Por último el cierre de esta primera iteración lo supuso la creación del primer prototipo funcional de la aplicación, que mediante un dispositivo móvil realizaba una llamada al servidor obteniendo como resultado la traducción de un texto palabra por palabra, de manera todavía poco precisa.

1.3.2. Iteración 2

Objetivos:

- ✓ Mejora funcional sustancial del software encargado de la traducción.
- ✓ Mejora sustancial del cliente Android.
- ✓ Ampliación de la memoria.

Desarrollo:

En esta segunda iteración nos centramos en la mejora funcional de la aplicación incluyendo un analizador que nos permite obtener información acerca de la frase a traducir y de este modo, conseguir una traducción mucho más refinada y no tan literal como la del primer prototipo. Este punto supuso además aprender a realizar llamadas a un servicio web desde otro servicio web y no desde un cliente en sí.

Por otro lado conseguimos avances en la parte del cliente Android, logrando mostrar los colores de los marcos de los pictogramas en función del tipo de palabra que sea, así como una importante mejora en eficiencia al descargar todas las imágenes de los pictogramas obtenidos en la traducción a la vez.

De igual modo y en paralelo se siguió trabajando en la memoria, añadiendo a la misma todos los conocimientos obtenidos, así como los desarrollos y avances producidos.

1.3.3. Iteración 3

Objetivos:

- ✓ Modificación de la arquitectura del servidor.
- ✓ Finalización del diseño del cliente Android.
- ✓ Finalización de la memoria.

Desarrollo:

Esta última iteración tuvo lugar a raíz de la decisión de modificar completamente toda la arquitectura del servicio web. Esta decisión nace de los



diversos problemas encontrados durante la realización del servidor web apoyado en Axis2, conocidos como servicios web "de tipo pesado". Estos problemas nos llevaron a afrontar el cambio a la arquitectura Rest que se comenta en mayor detalle en el capítulo 4, este tipo de servicio web conocidos como "ligeros", tras la implantación del mismo, nos permitió avanzar de forma más rápida y notablemente más sencilla en el desarrollo de la solución.

Asimismo, respecto al cliente Android, se consiguieron importantes progresos logrando mostrar una lista de todos los pictogramas disponibles obtenidos de la traducción para cada uno de los pictogramas mostrados como resultado, permitiendo así al usuario elegir el pictograma que más se adapte a lo que él buscaba.

Por último y al igual que en las iteraciones anteriores, se continuó trabajando en la memoria del proyecto, obteniendo la estructura final de la misma y reflejando en ella, a su vez, todos estos nuevos cambios en la arquitectura y los desarrollos del cliente para Android.



Capítulo 2:

Estado del arte

2. Estado del arte

2.1. Pictogramas

Un pictograma es un símbolo esquemático que representa una idea, concepto, etc. de forma que permita una comunicación superando las barreras presentes en el lenguaje.

El objetivo general de los pictogramas es el de informar o señalar. Sin embargo nosotros nos centraremos en su utilidad como Sistema Alternativo y Aumentativo de Comunicación (SAAC). En este ámbito los pictogramas se utilizan para comunicarse con personas que presentan dificultades en la comunicación ya sea oral o escrita.

Por tanto, para estas personas que presentan este tipo de dificultades, un pictograma debe suponer no solo una forma de expresar ideas o sentimientos, sino también una herramienta para interpretar, comprender y en definitiva transformar todo su entorno en imágenes de la forma más esquemática y eficiente posible para lograr de este modo una correcta comunicación con su interlocutor.

De este modo, los pictogramas pueden representar desde un objeto, animal, persona, etc.; a un sentimiento, una acción, e incluso elementos gramaticales.

A tal efecto hemos contactado con el Portal Aragonés ARASAAC (ARASAAC s.f.) para que nos proporcionara recursos gráficos y materiales que hiciesen más fácil la comunicación de personas con algún tipo de dificultad.

2.2. SAAC

Los Sistemas Aumentativos y Alternativos de Comunicación (SAAC) (ARASAAC s.f.) suponen una forma de comunicación distinta a las convencionales, ya sea hablando o por escrito, que persiguen facilitar lo máximo posible la comunicación a aquellas personas que presenten dificultades en este ámbito.

La comunicación y el lenguaje es, evidentemente, una necesidad básica para todo ser humano en cualquier actividad a realizar. Siendo imprescindible para aprender, relacionarse, divertirse o simplemente participar en la sociedad.

Gracias a los SAAC, todas las personas, independientemente de su edad, que carezcan de un nivel de comunicación suficiente para lograr hacerse entender de manera satisfactoria por el resto de la sociedad tienen a su disposición estos sistemas que permiten cubrir todas las necesidades comunicativas que puedan surgir.

Entre los motivos que pueden hacer necesario el uso de SAAC encontramos la discapacidad intelectual, la parálisis cerebral, trastorno autista (en sus diferentes espectros), enfermedades neurológicas, el párkinson o la esclerosis múltiple, los traumatismos cráneo-encefálicos, entre muchas otras.

La Comunicación Aumentativa y Alternativa (CAA) no surge como un sustituto del habla normal, si no que busca complementar las carencias que se puedan tener en ella y poco a poco permitir una rehabilitación del habla natural, permitiendo además que exista una comunicación cuando las dificultades en el habla no la hacen posible.

No supone por tanto un retraso en el desarrollo del habla natural si se introduce cualquiera de los Sistemas Aumentativos Alternativos existentes a edades tempranas, cuando se detecten dificultades en el desarrollo del lenguaje oral, ya sea por nacimiento, enfermedad o cualquier accidente que haya sufrido la persona. No existe prueba de que el uso de SAAC mitigue o interfiera de manera negativa en el desarrollo de la recuperación del habla natural.

2.2.1. Recursos que utilizan

La Comunicación Aumentativa y Alternativa incluye variedad sistemas de símbolos pudiendo englobarse en dos categorías, gráficos (como fotografías, dibujos, pictogramas, etc.) o gestuales (gestos o signos manuales). Además en el caso de los primeros, se requiere habitualmente el uso de productos de apoyo.

Los distintos tipos de símbolos se adaptan, según la edad, las habilidades motrices, cognitivas y lingüísticas, a las necesidades de las personas que requieren de su uso, conformando por tanto una amplia gama donde poder elegir en cada caso particular.

Es en este punto donde la tecnología juega un importante papel ya que los ordenadores personales, tablets u otros dispositivos con programas especiales desarrollados a tal efecto, permiten el acceso de forma adaptada, a una serie de herramientas con diferentes sistemas de signos pictográficos y ortográficos que facilitan en gran medida la comunicación a las personas que lo requieran.



Se incluyen también en este ámbito los recursos no estrictamente tecnológicos como pueden ser los tableros y libros de comunicación.

Para acceder a estas tecnologías existen diversos instrumentos, tales como los teclados y ratones adaptados o virtuales, los punteros, o los conmutadores que facilitan, a personas con capacidades limitadas en este aspecto, hacer uso de los recursos existentes.

En este punto cabe destacar que el software objetivo a desarrollar por el presente proyecto, está orientado a ser utilizado por personas que no presenten ningún tipo de discapacidad o dificultad en el ámbito de la comunicación, pero que deseen comunicarse con otras personas que sí las presenten y que por tanto les sea más sencillo comprender SAAC que texto convencional.

A tal efecto se ha creado este traductor de texto a pictogramas, que sin necesidad de que el usuario tenga conocimientos en el uso de los SAAC, le permita comunicarse de manera satisfactoria con personas que requieran de su uso.

2.2.2. Sistemas de símbolos

Las lenguas de signos que utilizan personas no oyentes no se consideran SAAC, ya que constituyen por sí mismos idiomas que se han desarrollado y se adquieren de forma natural, del mismo modo que ocurre con el lenguaje hablado convencional.

Existe una gran variedad de símbolos gráficos, desde sistemas sencillos basados en dibujos o fotografías hasta sistemas complejos como los sistemas pictográficos o la ortografía tradicional.

Por este motivo, gracias a los productos de apoyo y los distintos recursos para el acceso comentados anteriormente, los diferentes sistemas gráficos pueden ser utilizados no solo por personas con discapacidad intelectual, sino también por personas con discapacidades motoras, incluso en casos de extrema gravedad.

Los sistemas pictográficos utilizados el presente proyecto, se aplican perfectamente a personas que carecen de alfabetización a causa de la edad o la discapacidad. Con la ventaja adicional de permitir desde un nivel de comunicación elemental, que se adapta a personas con deficiencias graves, hasta un nivel de comunicación realmente rico.

21

Esta aplicación se caracteriza por su sencillez de uso que permite a cualquier persona crear y editar tableros. Además de permitir la comunicación mediante el uso de pictogramas (tomados de la base de datos de ARASAAC) la característica de realizar tableros es idónea para la elaboración y explicación de tareas rutinarias sencillas siendo de esta forma muy sencillo enseñar o planear con antelación infinidad de tareas para personas que presenten dificultad en la comunicación.



ILUSTRACIÓN 2. PANEL CREADO CON ARABOARD.

Además esta aplicación también presenta interesantes características, tales como permitir tomar fotografías mediante la cámara de nuestro dispositivo para crear con ellas pictogramas personalizados que podremos utilizar en los tableros que creamos.

2.3.2. AraWord

AraWord (AraWord s.f.) es un procesador de texto que permite la escritura simultánea tanto con texto como con pictogramas. Esta aplicación ha sido desarrollada teniendo como objetivo principal, la elaboración de materiales para personas que presenten dificultades en el ámbito de la comunicación. Es especialmente importante destacar que con la funcionalidad que ofrece la aplicación resulta sencillo realizar la adaptación de cualquier tipo de material que contenga texto convencional a texto con pictogramas, conformando así una excelente herramienta para la comunicación aumentativa y alternativa.

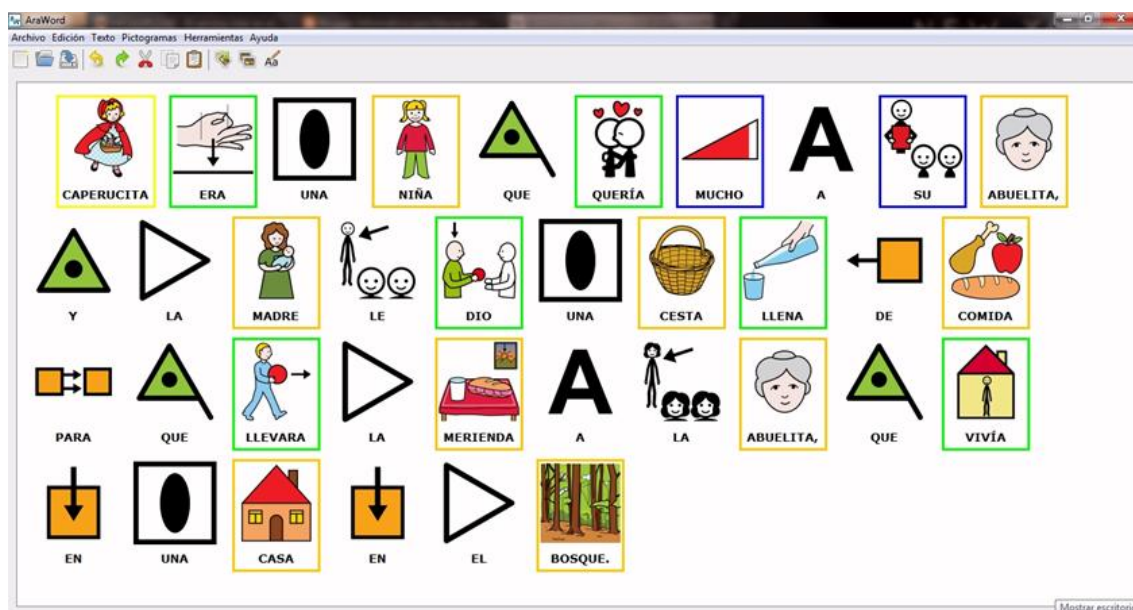


ILUSTRACIÓN 3. VISTA GENERAL DE LA APLICACIÓN ARAWORD.

2.3.3. TICO

Esta aplicación está compuesta a su vez por dos partes: el editor y el intérprete. Con el primero se permite diseñar tableros que contengan tango imágenes (pictogramas), sonidos o elementos de control del entorno como enlaces de navegación para visitar otros tableros contenidos en el mismo proyecto TICO.

Una vez creado un proyecto TICO es posible ejecutarlo con el intérprete de manera que se vayan sucediendo los tableros creados en un orden predefinido. Estas funcionalidades convierten a la aplicación en una importante herramienta para crear material útil y comprensible para personas que presentan dificultades en la comunicación y requieren del uso de sistemas aumentativos y alternativos de comunicación.

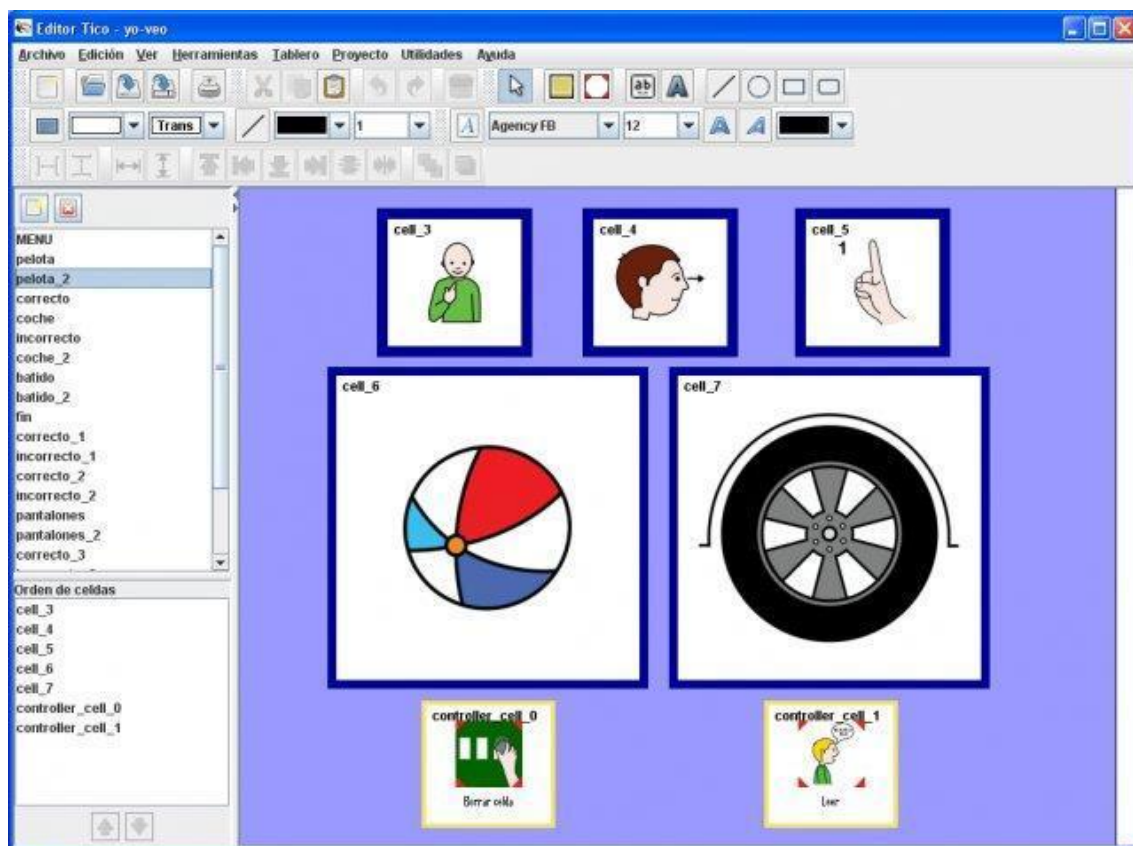


ILUSTRACIÓN 4. INTERFAZ DE LA HERRAMIENTA TICO.

2.3.4. Pictotraductor

La página web Pictotraductor (Pictoaplicaciones s.f.) es una herramienta pensada para facilitar la comunicación de padres y profesionales con personas que presenten dificultades en la expresión oral o escrita convencional y les resulte más sencillo el uso de imágenes para este fin.

El funcionamiento básico de esta página web consiste en introducir texto escrito en el área delimitada para ello y, a medida que se va escribiendo, se muestra la traducción de dicho texto a pictogramas.



ILUSTRACIÓN 5. INTERFAZ DE LA WEB PICTOTRADUCTOR.

Esta herramienta presenta destacables funcionalidades como la posibilidad de registrarse en la misma, pudiendo de este modo configurar la aplicación en parámetros como el tamaño de los pictogramas mostrados, crear imágenes propias a utilizar como pictogramas, guardar frases como favoritas o compartir la traducción en pictogramas con redes sociales o vía mail.

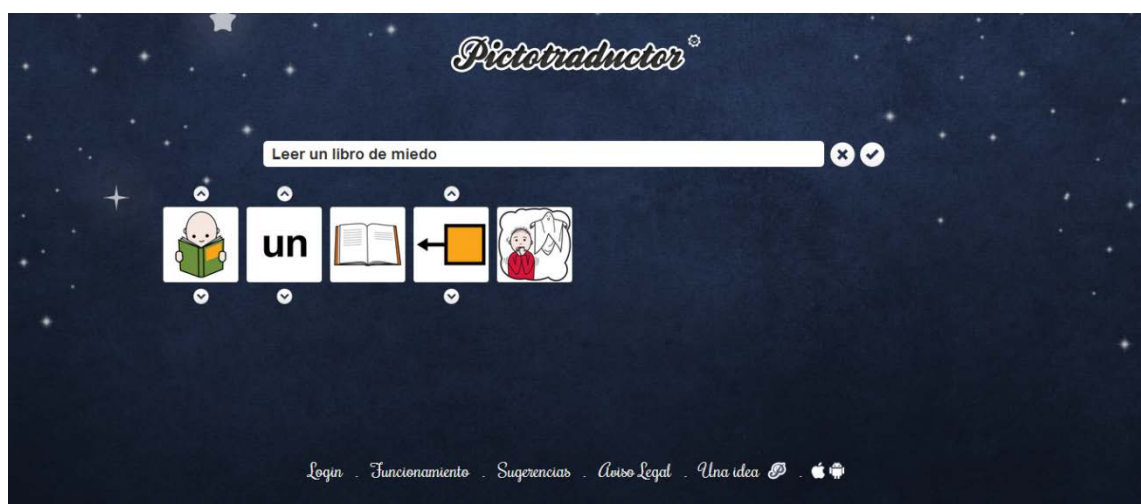


ILUSTRACIÓN 6. TRADUCCIÓN REALIZADA CON PICTOTRADUCTOR.

Cabe destacar que el presente proyecto busca una clara mejora de la traducción realizada con esta herramienta, proporcionando un análisis del texto a traducir que desemboca en la eliminación de determinantes haciendo de este modo que la traducción resultante sea mucho más sencilla de comprender por las personas a las que va destinada, así como la identificación del verbo y su infinitivo (en caso de que aparezca conjugado), la detección y correcta traducción de n-gramas y demás características que se detallan en esta memoria, que proporcionan una traducción más contextualizada y adaptada al objetivo real de la misma, personas con dificultades en el área de la comunicación.

2.4. Tecnología de servicios web

Como se ha comentado en el capítulo 1.3, esta parte del proyecto ha sufrido grandes cambios durante la ejecución del mismo, ya que, debido a los grandes problemas que encontramos en el desarrollo con la arquitectura planteada inicialmente, en la Iteración 3 decidimos abordar el cambio de la misma. Por este motivo a continuación se describe la arquitectura desarrollada en un principio y que posteriormente fue descartada, para pasar después a detallar la arquitectura finalmente implementada:

2.4.1. Tecnología inicial

Se comentan a continuación los aspectos, tecnologías y módulos más importantes que se utilizaron para desarrollar la arquitectura inicial del proyecto y que fue finalmente descartada:

2.4.1.1. Axis2

Apache Axis2 (Apache Software Foundation s.f.) es un motor utilizado para la creación de servicios web, el cual incluye soporte para varios estándares tales como:

ReliableMessaging, Coordination, AtomicTransaction, SecurityPolicy, Security, Trust, SecureConversation, SAML 1.1, SAML 2.0, WS - Addressing.

Aparte de la inclusión de los estándares antes mencionados, Axis2 ofrece estas otras características:

- Velocidad: Axis2 usa un modelo de objetos propio y un análisis sintáctico nuevo lo que desemboca en una velocidad de proceso mayor que la de versiones anteriores de Apache Axis.
- Ahorro de memoria: Axis2 ha sido diseñado pensando en generar la menor demanda de memoria posible.
- Sencillo despliegue: Axis2 permite realizar el despliegue del servicio copiando únicamente los archivos requeridos en el directorio de servicios del repositorio. Además permite añadir nueva funcionalidad al sistema sin tener que detener la ejecución del servidor.
- Servicios web asincrónicos: Axis2 soporta servicios web asincrónicos y realizar llamadas asincrónicas a los mismos.
- Soporte de WSDL: Axis2 soporta WSDL (*Web Services Description Language*).

2.4.1.2. SOAP

Sobre este motor se utiliza la tecnología SOAP (*Simple Object Access Protocol*) (W3C s.f.) el cual es un estándar que define la comunicación de dos objetos mediante el intercambio de datos XML.

SOAP es un *framework* de mensajería básica mediante el cual se pueden construir servicios web. Este protocolo cuenta con tres partes: un sobre (*envelope*), en el que se define qué hay en el mensaje y cómo se debe procesar. El protocolo SOAP tiene las siguientes características principales:

- Es neutro: se puede utilizar sobre cualquier protocolo de transporte.
- Es independiente: permite la utilización de cualquier modelo de programación.

2.4.1.3. WSDL

WSDL (*Web Services Description Language*) (W3C s.f.), se utiliza para describir servicios web.

WSDL está basado en XML y describe la forma en la que el servicio web se comunica, incluyendo los requisitos de los protocolos implicados y los formatos de los mensajes utilizados.



De este modo el cliente puede leer el WSDL del servicio web y saber así que funciones le facilita el servidor. En conclusión el WSDL nos proporciona información sobre la funcionalidad del servicio web y como utilizarla.

2.4.1.4. Tomcat

Cabe destacar también la utilización de Tomcat (Apache Software Foundation s.f.) durante el desarrollo de la aplicación del servidor web con soporte de *servlets* y JSPs Tomcat, permitiendo realizar los desarrollos y pruebas pertinentes de la parte del servidor y del servicio web objeto del presente proyecto.

2.4.1.5. ksoap2

Una vez definida la arquitectura de la parte del servidor, nos centramos en este punto en la conexión del cliente Android al servicio web. Esta conexión se realiza mediante el uso de ksoap2 (Ksoap2 Android s.f.) que proporciona una librería cliente SOAP, ligera y eficiente para la plataforma Android.

2.4.2. Tecnología final

La aplicación está dividida en dos partes bien diferenciadas; la parte cliente diseñada e implementada en Android y la parte del servidor que se encarga de realizar todas las operaciones y de devolver el resultado al cliente.

Para realizar la parte del servidor hemos decidido crear un servicio web REST que sea el que se encargue de recibir la petición del cliente Android, consultar la base de datos para obtener el pictograma y devolver el resultado correctamente.

2.4.2.1. REST

REST (*Representational State Transfer*) es un modelo de arquitectura de desarrollo web creado totalmente sobre el estándar HTTP. Nos permite crear servicios y aplicaciones que puedan ser usadas por cualquier dispositivo que entienda HTTP (prácticamente cualquier dispositivo moderno) por lo que es mucho más simple que otras alternativas que consideramos como SOAP. REST se caracteriza por:

- Ser un protocolo cliente/servidor sin estado, es decir, cada comunicación entre el cliente y el servidor se realiza de forma independiente sin necesidad de guardar ninguna información entre ellas.
- Tener un conjunto de operaciones claras y bien definidas que se asemejan mucho a las operaciones CRUD en bases de datos. Las más utilizadas son POST, GET, PUT, DELETE.

Podemos encontrar en la web multitud de recursos REST. Entre ellos cabe destacar que grandes empresas e industrias utilizan esta tecnología como por ejemplo Twitter, MEGA, Facebook, Amazon, etc.

Para facilitar la construcción de nuestro servicio REST utilizamos el framework Spring, que nos permite crear de una forma sencilla y un poco particular este tipo de aplicaciones. Además utilizaremos otro framework llamado Hibernate y aplicaremos el patrón DAO para mejorar el acceso a datos y gestionar de forma más eficiente la conexión con la base de datos. Vamos a explicar brevemente cada uno ellos.

2.4.2.2. Spring MVC

Spring (Pivotal Software s.f.) es un contenedor de aplicaciones desarrollado para Java, que proporciona características y funcionalidades útiles como la inversión de control, la inyección de dependencias, la gestión de transacciones y el acceso a datos abstractos entre otros. Surgió en 2002 de la mano de Rod Johnson, pero su lanzamiento más importante fue en diciembre de 2013 ya que incluye soporte para las herramientas más actuales y es totalmente compatible con Java EE 7, Java SE 8...

Spring Framework se ha popularizado entre la comunidad de programadores en Java como alternativa del modelo de Enterprise JavaBean ya que, además de no obligar a utilizar únicamente un modelo de programación, es capaz de ejecutar cualquier EE Java incluido el servidor. Además de esta libertad a la hora de programar, ya existen soluciones a problemas bien documentadas y fáciles de usar, y se han creado multitud de extensiones y mejoras para crear aplicaciones basadas en web.

Spring está basado en los siguientes principios:

- El buen diseño es más importante que la tecnología subyacente.
- Los *JavaBeans* deben estar unidos a las interfaces de una forma más independiente que hasta ahora.
- El código debe ser fácil de probar.

Spring consta de más de veinte módulos que tienen diferente funcionalidad y que se pueden combinar entre ellos para conseguir el resultado elegido. Para nosotros los módulos más importantes son:

Spring MVC:

Es una adaptación un poco especial del clásico Modelo-Vista-Controlador. Es un patrón de arquitectura de las aplicaciones software que separa por un lado los datos y la lógica de negocio, y por el otro lado la interfaz de usuario y el controlador encargado de gestionar los eventos y las comunicaciones. En nuestro caso tenemos el modelo que representa la base de datos, tenemos los controladores encargados de atender las peticiones del usuario, y por último la vista sería la interfaz gráfica con la que interactúa el usuario, que sería nuestra aplicación Android.

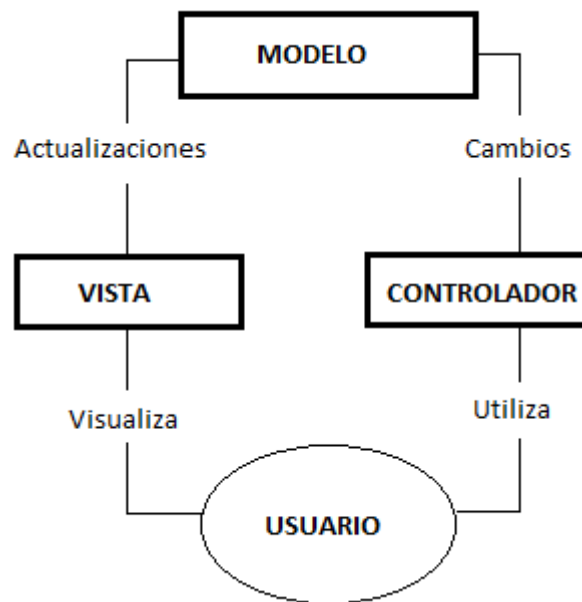


ILUSTRACIÓN 7. DIAGRAMA DEL FRAMEWORK SPRING.

Inversión de control:

Se encarga principalmente de la inyección de dependencias (Ruiz Pacheco s.f.). En nuestro caso se encarga de inyectar las dependencias necesarias, indicadas mediante anotaciones, en lugar de instanciarlas en el momento. Un ejemplo básico de esta funcionalidad es el uso de `@Autowired` para instanciar las clases que representan la base de datos y poder trabajar con ellas sin tenerlas que crear de nuevo. Se puede llevar a cabo a través de los métodos mutadores o por la propia constructora.

Acceso a datos:

Utiliza el patrón DAO junto con los conectores a bases de datos como JDBC o con un ORM entre los que se encuentra Hibernate que es el que utilizamos nosotros.

2.4.2.3. Hibernate

Hibernate (Redhat s.f.) es un framework de persistencia de datos a una base de datos (también denominado ORM que significa *"Object-Relational mapping"*). Como todas las herramientas de su tipo, Hibernate intenta solucionar el problema de las diferencias entre los dos modelos de datos existentes al mismo tiempo en una aplicación: el usado en la memoria del servidor (orientado a objetos) y el usado en las bases de datos (modelo relacional). Además incluye un concepto muy importante, las



sesiones. Las sesiones son las conexiones que se establecen con la base de datos cuando queremos acceder a los datos, Hibernate maneja estas conexiones evitando que se creen nuevas cada vez que realizamos una consulta, asignando en caso de que ya haya creadas y estén sin utilizar, lo que mejora el rendimiento y evita errores comunes a la hora de hacer consultas simplemente usando JDBC como no cerrar una sesión abierta.

2.4.2.4. Patrón DAO

DAO (Data Access Object) (Oracle s.f.) es un patrón de diseño de software que se encarga de aislar la aplicación del gestor de bases de datos utilizado para almacenar la información.

Es importante utilizar este patrón si queremos que los componentes de nuestra aplicación sean independientes del sistema de persistencia de datos, para poder permitir distintos tipos de almacenamiento, migraciones entre distintos fabricantes y diferentes fuentes de datos. El patrón DAO se encarga de resolver este problema usando un objeto llamado Objeto de Acceso a Datos que abstrae y encapsula el acceso a los datos. Además siempre realiza operaciones atómicas contra la base de datos para manejar la conexión y guardar los datos.

Para ello, normalmente se crea una clase para gestionar las conexiones y el transporte de los datos, independiente de la clase que contiene la información, de esta forma un cambio en la gestión como un cambio de tipo de almacenamiento no afectará a los datos porque seguirán guardándose con la misma estructura.



Capítulo 3: Herramientas utilizadas

3. Herramientas utilizadas

3.1. Android

Android (Android s.f.) es un sistema operativo pensado para dispositivos móviles basados en pantalla táctil. Este sistema operativo está basado en Linux y es actualmente el sistema operativo más extendido en este tipo de dispositivos.

3.1.1. Componentes principales

Los principales componentes y características de Android son:

- Aplicaciones base: incluyen una amplia gama de funcionalidades básicas tales como un cliente de correo electrónico, SMS, calendario, mapas, navegador, contactos, etc.
- *Framework* de aplicaciones: los desarrolladores cuentan con acceso completo a los mismos APIs del *framework* usados por las aplicaciones base. La arquitectura está diseñada para permitir la reutilización total de cualquiera de sus componentes, es decir, cualquier aplicación puede ser publicada y cualquier otra aplicación puede luego hacer uso de sus funcionalidades (sujeto a ciertas reglas evidentemente).
- Bibliotecas: el sistema operativo incluye bibliotecas de C/C++. Estas características también están disponibles para los desarrolladores.
- *Runtime* de Android: cada aplicación Android cuenta con su propio proceso optimizado.
- Basado en Linux: Android está basado en Linux en varios aspectos tales como los servicios de seguridad, gestión de memoria, gestión de procesos, etc.

3.2. Base de Datos ARASAAC

Para poder realizar la traducción de texto a pictogramas es imprescindible tener una base de datos donde poder almacenar todas las palabras que tienen asociado un pictograma. Para ello nos pusimos en contacto con ARASAAC que nos proporcionó todos los pictogramas de los que disponía la colección hasta el momento, aproximadamente 8500 imágenes identificada cada una de ellas por un identificador numérico. Estos pictogramas están catalogados a su vez en un archivo Excel, que

asocia a cada imagen sus palabras correspondientes, en total encontramos unas 15000 palabras aproximadamente.

En este archivo Excel se almacena toda la información de cada pictograma. Cada fila contiene la siguiente información:

En la primera columna encontramos el nombre de la imagen y en las siguientes columnas encontramos una lista con todas las palabras que se corresponden con esa imagen en pares <palabra=color>. Todos los pictogramas están traducidos a español, inglés, alemán, italiano, francés, euskera y catalán, pero nosotros solo nos quedaremos con las palabras en español.

Los colores asociados a cada palabra las identifican de la siguiente manera:

- nombres propios - amarillo.
- nombres comunes - naranja.
- acciones - verde.
- descriptivos - azul.
- social - rosa.
- miscelánea - blanco (sin marco).
- no encontrado en la base de datos - violeta (añadido por nosotros).

En la Ilustración 8 podemos ver un pequeño fragmento de este archivo. Si analizamos la fila dos podemos ver que el pictograma 2239 hace referencia a la palabra abeja y que su color es el 2 que hace referencia a que es un sustantivo. Y en la Ilustración 9 podemos ver el pictograma correspondiente, o uno de ellos, porque a veces para una misma palabra existen varios pictogramas asociados.

| | A | B | C |
|----|-----------|--|--------------------------------------|
| 1 | ID imagen | Castellano | Inglés |
| 2 | 2239 | abeja=2; | bee=2; |
| 3 | 2242 | vestido=2;abrigo=2; | dress=2;coat=2; |
| 4 | 2243 | abuela=2;yaya=2;abuelita=2; | grandmother=2;granny=2;grandma=2; |
| 5 | 2244 | abuelo=2;yayo=2;abuelito=2; | grandfather=2;grandpa=2;grandpa=2; |
| 6 | 2245 | aburrido=4;aburrido=4;aburrir=3; | boring=4;bore=4;to get tired=3; |
| 7 | 2246 | aceite=2;aderezo=2;aceitera=4;aceite de oliva=2; | oil=2;seasoning=2;oil=4;olive oil=2; |
| 8 | 2247 | acera=2; | pavement=2; |
| 9 | 2248 | agua=2;bebida=2; | water=2;drink=2; |
| 10 | 2249 | alfombra=2; | carpet=2; |
| 11 | 2250 | almohada=2; | pillow=2; |
| 12 | 2251 | ambulancia=2; | ambulance=2; |

ILUSTRACIÓN 8. EJEMPLO DEL ARCHIVO EXCEL PROPORCIONADO POR ARASAAC.



ILUSTRACIÓN 9. EJEMPLO DE PICTOGRAMA ARASAAC, IMAGEN 2239 (ABEJA).

3.3. Maltparser

Maltparser (Johan Hall s.f.) es un generador de analizadores de dependencias contemporáneo basado en aprendizaje automático mediante el que se obtiene una gran precisión (Ballesteros 2010).

El sistema realiza un análisis gramatical que permite que cada frase de un texto sea analizada, es decir, anotada junto a su estructura gramatical, lo que encaja a la perfección con la necesidad presente en nuestro proyecto de realizar una traducción realmente fiel y precisa de texto.

De este modo, Maltparser nos ofrece el análisis gramatical del texto que deseemos devolviéndonoslo en un archivo .txt como el que se muestra en la Ilustración 10.

| | | | | | | | | | |
|---|------------|-----------|---|---------|---|---|------|---|---|
| 1 | mi | mi | d | DP1CSS | _ | 2 | SPEC | _ | _ |
| 2 | madre | madre | n | NCFS000 | _ | 3 | SUBJ | _ | _ |
| 3 | come | comer | v | VMIP3S0 | _ | 0 | ROOT | _ | _ |
| 4 | magdalenas | magdalena | n | NCFP000 | _ | 3 | DO | _ | _ |
| 5 | por | por | s | SPS00 | _ | 3 | MOD | _ | _ |
| 6 | la | el | d | DA0FS0 | _ | 7 | SPEC | _ | _ |
| 7 | noche | noche | n | NCFS000 | _ | 5 | COMP | _ | _ |

ILUSTRACIÓN 10. EJEMPLO DEL ARCHIVO .TXT DEVUELTO POR MALTPARSER.



Como se puede observar en la ilustración anterior, el archivo incluye por cada columna información útil para cada palabra, tal como la palabra en sí y su raíz (si es que la tiene), si es un determinante, un nombre, un verbo, las etiquetas gramaticales y las dependencias entre las palabras que forman la frase.

Esta funcionalidad está disponible en un servicio web por lo que, en nuestro caso, realizamos una llamada desde nuestro servicio web a este servicio, pasándole como argumento la frase que llega desde el cliente.

Es, principalmente, por ser una herramienta tan accesible, al estar disponible como servicio web, por lo que hemos escogido este analizador gramatical para la realización del proyecto.



Capítulo 4:

Arquitectura del

sistema

4. Arquitectura del sistema

En este capítulo realizamos un primer acercamiento a la arquitectura desarrollada dentro del proyecto, para pasar a comentar cada una de sus partes diferenciadas con mucho mayor nivel de detalle en capítulos posteriores.

4.1. Descripción general de la arquitectura

La arquitectura cuenta con varias partes bien diferenciadas, como se muestra en la Ilustración 11.

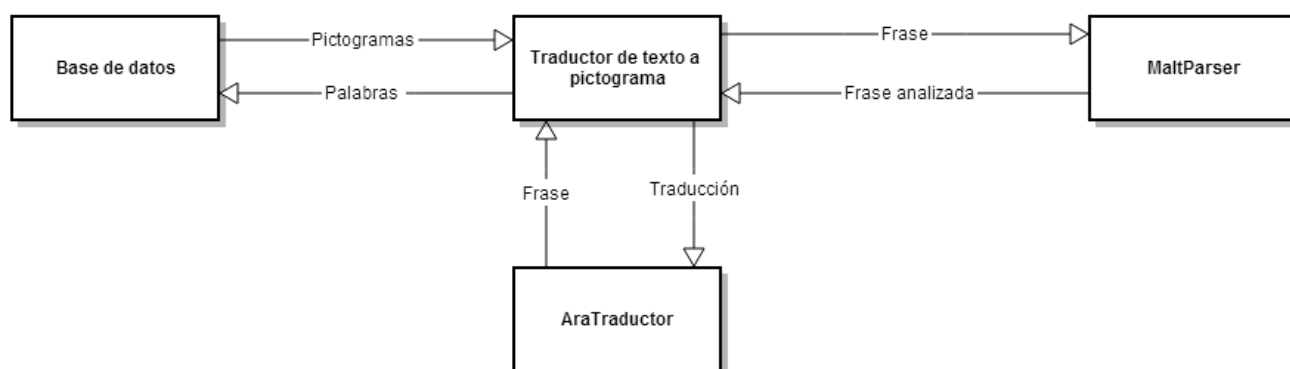


ILUSTRACIÓN 11. DIAGRAMA DE ALTO NIVEL DE LA ARQUITECTURA.

Primero el usuario introduce la frase que desea traducir en el cliente Android, el cual sustituye espacios y saltos de línea por comas para su correcto envío al Traductor de texto a pictograma. Este realiza un pre-procesamiento que consiste en preparar la frase para su envío a Maltparser (tratamiento de "ñ", acentos, comas, etc.). En definitiva, recupera los caracteres especiales que el protocolo http no reconoce. Posteriormente, envía la frase al servicio web Maltparser que realiza un análisis de la misma y la devuelve al Traductor de texto a pictograma. Con dicho análisis se realizan las consultas pertinentes a la base de datos que nos devuelve los pictogramas correspondientes a cada palabra y se envían al cliente para que muestre la traducción.

4.1.1. Base de Datos

Desarrollada por nosotros, contiene la información necesaria para devolver los pictogramas asociados a cada palabra.

4.1.2. Traductor de texto a pictogramas

Es la parte funcional del proyecto, encargada de realizar la traducción en sí de texto a pictogramas. Desarrollado por nosotros, Se encuentra alojado en el servidor y recibe del cliente Android, AraTraductor, la frase a traducir, con ella se realiza una llamada a Maltparser para analizarla. Posteriormente se analiza la existencia de n-gramas dentro de la frase y por último se realizan las consultas necesarias a la base de datos para devolver al cliente la traducción deseada.

4.1.3. AraTraductor

También desarrollado por nosotros, es el cliente que consume las funcionalidades del servicio web. Es la parte del proyecto visible por el usuario. Se debe encontrar instalado en el dispositivo móvil que se desee y mediante él se introduce la frase a traducir y se muestra el resultado.

4.1.4. Maltparser

Servicio web ya existente que nos permite, mediante una llamada con la frase deseada, realizar el análisis léxico y sintáctico de la misma.

4.2. Implementación de la arquitectura

Se comenta en este punto la implementación llevada a cabo por cada uno de los módulos comentados anteriormente:

4.2.1. Base de Datos

Implementada mediante MySQL, el traductor de texto a pictogramas le realiza una consulta por cada palabra o palabras del texto devolviéndole a su vez la url asociada. Su estructura se comenta en detalle en el capítulo 5.

4.2.2. Traductor de texto a pictogramas

Implementado como servicio web REST, en el lenguaje de programación JAVA, recibe del cliente la frase a traducir mediante un *string* y tras llevar a cabo un primer pre-procesamiento realiza, mediante un archivo .jar la llamada al servicio web de Maltparser con la frase procesada como argumento, el cual le devuelve un archivo .txt con la información gramatical de la frase. Se comenta en profundidad en el capítulo 6.

4.2.3. AraTraductor

Desarrollado como aplicación Android, envía como *string* el texto a traducir al traductor de texto a pictogramas y recibe de este un objeto de la clase Json que trata adecuadamente para mostrar la traducción final. Se comentan todas sus características en el capítulo 7.

Se incluyen por tanto en la Ilustración 12 y en la Ilustración 13, un diagrama de la arquitectura con mayor grado de detalle en los flujos de información y un ejemplo del archivo .txt que nos devuelve el servicio web de Maltparser, respectivamente.

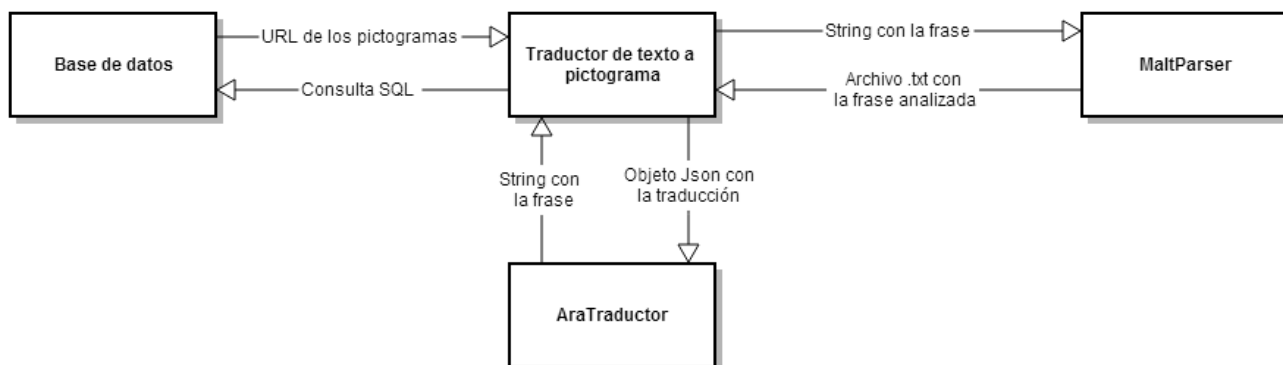


ILUSTRACIÓN 12. ARQUITECTURA CON MAYOR GRADO DE DETALLE.

En este nuevo diagrama se muestra de forma más técnica los distintos flujos de información presentes en la arquitectura. El cliente Android envía la frase a traducir mediante un *string* al Traductor de texto a pictograma, el cual la envía, también mediante un *string*, al servicio web de Maltparser. Éste devuelve un archivo .txt con el análisis gramatical de la frase a nuestro servicio web, del que extraemos la información oportuna para realizar las consultas HQL (consultas SQL con el patrón



Hibernate) a la base de datos. Finalmente el Traductor de texto a pictograma construye un objeto Json con las URLs que devuelve la base de datos y envía dicho objeto al cliente Android.

| | | | | | | | | | |
|---|------------|-----------|---|---------|---|---|------|---|---|
| 1 | mi | mi | d | DP1CSS | _ | 2 | SPEC | _ | _ |
| 2 | madre | madre | n | NCFS000 | _ | 3 | SUBJ | _ | _ |
| 3 | come | comer | v | VMIP3S0 | _ | 0 | ROOT | _ | _ |
| 4 | magdalenas | magdalena | n | NCFP000 | _ | 3 | DO | _ | _ |
| 5 | por | por | s | SPS00 | _ | 3 | MOD | _ | _ |
| 6 | la | el | d | DA0FS0 | _ | 7 | SPEC | _ | _ |
| 7 | noche | noche | n | NCFS000 | _ | 5 | COMP | _ | _ |

ILUSTRACIÓN 13. EJEMPLO DEL .TXT DEVUELTO POR MALTPARSER.



Capítulo 5:

Base de datos

5. Base de datos

A partir de los recursos facilitados por ARASAAC, comentados en el punto 3.2, decidimos generar nuestra propia base de datos. Para ello utilizamos una base de datos relacional MYSQL, que configuramos según nuestras necesidades.

Entre las distintas posibilidades de diseño de la base de datos decidimos crear dos tablas para almacenar la información, una para las palabras y otra para los pictogramas. Nos decantamos por esta solución porque a pesar de que para el desarrollo de nuestra aplicación no necesitamos buscar en las tablas mediante el pictograma en otro momento si podría ser de gran ayuda. La principal función de la base de datos es devolver todos los pictogramas que están asociados a una palabra, para ello sólo es necesario la ejecución de una consulta.

La tabla PALABRAS almacena las palabras que tienen pictograma, para ello creamos los siguientes campos:

- ID: Identifica de forma unívoca cada palabra, es un entero auto incrementable usado como clave de la tabla .
- nombre: Contiene la palabra en cuestión.
- id_url: Identifica el ID de la imagen que representa la palabra.
- color: Indica el color del pictograma, depende del tipo de palabra y como ya se ha indicado anteriormente relaciona palabra y tipo de la siguiente forma:
 - nombres propios - amarillo.
 - nombres comunes - naranja.
 - acciones - verde.
 - descriptivos - azul.
 - social - rosa.
 - miscelánea - blanco (sin marco).
 - no encontrado en la base de datos - violeta (añadido por nosotros).

5.1. Descripción de las tablas

La tabla PICTOGRAMAS almacena los distintos pictogramas con la dirección URL necesaria para obtener el recurso. Para ello consta de los siguientes campos:

- **id_pictograma:** Identifica de forma unívoca cada pictograma y es el elemento común con la tabla PALABRAS siendo el nombre del archivo que contiene la imagen que representa. Además es la clave de la tabla.
- **url:** dirección web donde encontraremos la imagen. Este campo apunta a una dirección con el siguiente formato http://hypatia.fdi.ucm.es/conversor/Pictos/****.png , siendo los asteriscos el **Id_pictograma**.

5.2. Creación de la base de datos en formato SQL

Crear la base de datos copiando los datos a mano no tenía mucho sentido, ya que nos encontramos con aproximadamente 15000 palabras referenciadas y 8500 imágenes. Por lo que decidimos implementar una función en Java que automáticamente hiciera ese trabajo por nosotros.

La función consiste en recorrer una a una las filas de la tabla Excel y tratar cada uno de los campos, el primer campo que se recoge es el *ID_Imagen* que se corresponde con nuestro *id_Pictograma* en la tabla PICTOGRAMAS y con el *id_url* en la tabla de PALABRAS. En la segunda columna encontramos la lista de palabras que hacen referencia a esa imagen junto con su color, de manera que hay que separarlos para guardar en nuestra base de datos. Veamos un ejemplo:

- **Excel:** Leemos una fila del archivo Excel donde vemos que el primer dato es el nombre de la imagen a la que hacen referencia todas las palabras que aparecen a continuación (es el nombre de la imagen que podemos encontrar en el archivo .zip que ARASAAC nos envió con todas las imágenes). La lista de palabras está formada por una palabra y un número, siendo el número el color que esa palabra tiene asociado.

| | |
|------|--|
| 2246 | aceite=2;aderezo=2;aceitera=4;aceite de oliva=2; |
|------|--|

ILUSTRACIÓN 14. FILA DEL EXCEL PROPORCIONADO POR ARASAAC A TRANSFORMAR.

- **Base de datos:** Guardamos la información de cada fila en la base de datos creando dos tablas la primera que guarde el nombre de la imagen con su dirección url de acceso al recurso, y en la otra tabla cada una de las palabras

con el color correspondiente y con el nombre de la imagen para hacer referencia a la otra tabla.

| id_pictograma | url |
|---------------|---|
| 2246 | http://hypatia.fdi.ucm.es/conversor/Pictos/2246.png |

ILUSTRACIÓN 15. TRANSFORMACIÓN OBTENIDA EN LA TABLA **PICTOGRAMAS** TRAS LA EJECUCIÓN DE NUESTRA FUNCIÓN JAVA.

| id_palabra | nombre | id_url | color |
|------------|-----------------|--------|-------|
| 12 | aceite | 2246 | 2 |
| 15 | aceite de oliva | 2246 | 2 |
| 14 | aceitera | 2246 | 4 |
| 13 | aderezo | 2246 | 2 |

ILUSTRACIÓN 16. TRANSFORMACIÓN OBTENIDA EN LA TABLA **PALABRAS** TRAS LA EJECUCIÓN DE NUESTRA FUNCIÓN JAVA.

Además de crear la base de datos solucionamos algunos problemas como la repetición de las mismas palabras referenciando a una imagen, lo que provocaría que se devolvieran valores repetidos. En la Ilustración 17, vemos un ejemplo del problema que estamos mencionando, donde aparece varias veces la palabra "bañador" haciendo referencia siempre a la misma imagen.

| | |
|------|--------------------------------|
| 2270 | bañador=2;bañador=2;bañador=2; |
|------|--------------------------------|

ILUSTRACIÓN 17. REPETICIÓN DE DATOS



Capítulo 6:

Traductor de texto a

pictogramas

6. Traductor de texto a pictogramas

En este capítulo describimos en primer lugar la funcionalidad del traductor en sí, para posteriormente, pasar a detallar con mayor detalle como se ha implementado dicha funcionalidad.

6.1. Funcionalidad

Vamos a describir los pasos que sigue una petición a nuestro servicio REST. Una vez que se ha invocado a nuestro servicio web mediante la dirección URL <http://hypatia.fdi.ucm.es:5223/PictoTranslatorApplicationServer/api/TraduceFrase/frase> siendo frase las palabras que se quieran traducir a pictogramas. Cuando el controlador descrito recibe la petición llama a la función encargada de tratar la petición y de realizar la consulta, pero antes se debe realizar un pre-procesamiento de la frase para adecuarla a nuestro formato, es necesario:

Corregir los errores de ortografía ya que las tildes, la 'ñ', y los signos de interrogación y exclamación se transforman en otros caracteres totalmente distintos. Por ejemplo para la palabra "él" llegaría la palabra "Ã©l", lo cual provocaría un error.

Una vez recorrida toda la frase y almacenada la solución la devolvemos mediante un archivo Json, que el cliente recibe y se encarga de tratar. El archivo devuelto tiene la estructura que se muestra en la Ilustración 18.

```
[  
  {  
    "id_palabra": 461,  
    "nombre": "perro",  
    "id_url": 2517,  
    "color": 2  
  },  
  {  
    "id_palabra": 4534,  
    "nombre": "perro",  
    "id_url": 7202,  
    "color": 2  
  }  
]
```

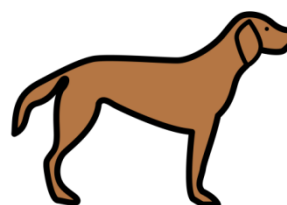
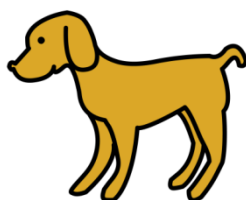


ILUSTRACIÓN 18. EJEMPLO DEL RESULTADO DEVUELTO Y SUS PICTOGRAMAS ASOCIADOS.

Se trata de una lista de palabras que contiene en cada elemento la lista de las imágenes asociadas a esa palabra. En este ejemplo se puede ver que la palabra

"perro" tiene dos imágenes asociadas, estas imágenes se podrán encontrar en nuestro servidor.

6.2. Implementación

En este apartado se describe la implementación realizada para la funcionalidad comentada anteriormente.

Invocar al servicio web de Maltparser que se ha descrito anteriormente para que analice sintáctica y morfológicamente la frase. El resultado será devuelto en un archivo de texto con el que trabajaremos.

Una vez tenemos el resultado de Maltparser leemos la información y realizamos la búsqueda de los pictogramas. Ya hemos explicado que hay gran cantidad de pictogramas que hacen referencia a expresiones que tienen más de una palabra por lo tanto debemos buscar n-gramas. Decidimos no buscar más de 3-gramas porque su número es muy reducido y el tiempo que se tarda en buscarlos no compensa. Por lo tanto buscamos todos los n-gramas posibles en la frase y comprobando que no tienen un pictograma asociado en la base de datos, si no se encuentra el n-grama pasamos a buscar la palabra según nos llega del cliente y en caso de que tampoco se encuentre buscamos la palabra simple de la que procede, valor que nos ha devuelto Maltparser.

6.2.1. Tratamiento de n-gramas

En este punto cabe destacar un procedimiento especial que realiza nuestra herramienta para un tipo específico de palabras.

Estas son los n-gramas, denominadas así por consistir en una secuencia de n palabras con un significado único en conjunto. De este modo, una palabra sola se le considera un 1-grama o unigrama, una compuesta de dos palabras 2-grama, bigrama o digrama (por ejemplo "crema solar") y así sucesivamente.

Es evidente que tratar de realizar un traductor sin tener en cuenta este tipo construcción léxica no tiene sentido, es por ello que se ha desarrollado un método para tenerlas en cuenta en nuestra traducción a pictogramas.

Este método consiste fundamentalmente en, dada la frase a traducir como parámetro de entrada, ir cogiendo las distintas agrupaciones (2-grama, 3-grama, etc.)

sucesivamente y realizar con ellas una consulta directa en la base de datos para ver si tienen un significado propio como conjunto, de ser así se marcan esas palabras como pertenecientes a un n-grama y por lo tanto el pictograma que se ofrece como traducción es el del n-grama correspondiente y no los pictogramas de las palabras individuales.

Es importante destacar que en nuestro caso únicamente contemplamos la existencia de hasta 3-gramas, ya que es difícil encontrar n-gramas compuestos por más de tres palabras y de este modo obtenemos un límite razonable tanto en resultado óptimo como en eficiencia para nuestra búsqueda.

Además añadimos otra peculiaridad y es que un n-grama muchas veces viene formado por un verbo, y en estos casos no podemos buscar en la base de datos uniendo simplemente las palabras sino que tenemos que hacer la consulta con la palabra simplificada devuelta por el analizador MaltParser. De esta forma decidimos buscar los n-gramas de todas las formas posibles, combinando palabras simplificadas con palabras que no lo están. Para explicar esto, utilizaremos un ejemplo muy sencillo:

1. Recibimos la frase de entrada: **Mi hijo fue de excursión a la montaña.**
2. Buscamos todos los pictogramas sin buscar n-gramas y encontramos siete pictogramas, pero vamos a centrarnos en la parte que realmente nos interesa en este apartado que es **fue de excursión**, el resultado de estas tres palabras sería el representado en la Ilustración 19.

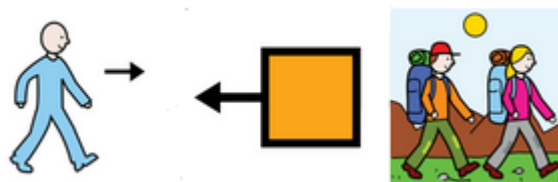


ILUSTRACIÓN 19. FRASE SIN N-GRAMAS.

3. Ahora buscamos la frase con n-gramas realizando todas las combinaciones posibles de grupo de tres y dos palabras y vemos que la frase "**fue de excursión**" tal cual no es un n-grama, pero y si buscáramos utilizando el lema de "**fue**", que es el verbo ir, encontraríamos lo mostrado en la Ilustración 20.



ILUSTRACIÓN 20. PICTOGRAMA DEL N-GRAMA "IR DE EXCURSIÓN".

De esta forma sólo utilizamos una imagen para representar tres palabras, lo que hace que quede más simplificado y por lo tanto más fácil de entender.

6.3. Implementación del traductor como servicio web

Se describe en este punto la estructura de la parte del servidor desarrollada.

Debido a la utilización de diferentes *frameworks* y patrones la estructura viene en cierto modo impuesta por estos. Por ello hemos decidido dividirla en dos partes muy diferenciadas. La primera se ocupa de la parte de persistencia de datos, los principales módulos ocupados de esta función son el patrón DAO e Hibernate. Como hemos descrito antes es importante separar el modelo de datos para que en caso de necesitar realizar un cambio de base de datos no tener que cambiar toda la aplicación.



ILUSTRACIÓN 21. ESTRUCTURA DEL SERVIDOR DESARROLLADO.

Como podemos observar en la Ilustración 21, existe un paquete llamado *persistence*, dentro de él aplicaremos el patrón DAO generando la interfaz *GenericDAO* y la clase *GenericDAOImplHibernate* que se encarga de implementar esta interfaz y como su propio nombre indica utiliza Hibernate para la persistencia de los datos. En caso de querer cambiar el modelo de persistencia por ejemplo para aplicar el protocolo de persistencia JPA solo sería necesario modificar esta clase.

Además en el paquete *persistence* también encontramos el paquete Hibernate que se encargará de gestionar la conexión con la base de datos mediante las sesiones.

La otra parte diferenciada de la aplicación es la creación del servicio REST. Para ello como hemos explicado antes utilizamos el framework de Spring basado en el modelo vista controlador (MVC). Hay tres paquetes importantes en esta parte: *domain*, *api* y *dao*.

6.3.1. *Com.ssiipicto.server.domain*

En este paquete se encuentran las clases que representan objetos de java. Nuestros objetos son las dos tablas que hemos descrito en la base de datos, junto con los datos que se devuelven una vez que el cliente invoca el servicio REST.

Las clases Palabras y Pictogramas sirven para generar los objetos que Hibernate se encarga de hacer persistir con la base de datos. Tienen exactamente la misma estructura que tienen las tablas de la base de datos, y mediante anotaciones de Spring podemos crear la relación entre ambas.

Si tomamos como ejemplo la clase Palabras nos encontramos las siguientes anotaciones:

@Entity: Indica que la clase está asociada a una entidad (relacionada con una base de datos).

@Table (name = "palabras"): Nos indica la tabla de la base de datos a la que hace referencia, en este caso hace referencia a la tabla Palabras.

Una vez descrita la clase pasamos a describir sus atributos, los atributos para la clase Palabras son como las columnas de la tabla Palabras, por lo tanto tendrá que haber tantos atributos como columnas tiene la tabla, y mediante las anotaciones correctas podremos indicar la columna a la que hace referencia (**@Column(name = "id_palabra")**), también es importante añadir a cada atributo las mismas características que tiene en la base de datos, es decir, es necesario indicar si es clave primaria o si es auto-incrementable.

6.3.2. *Com.ssiipicto.server.api*

En este paquete encontramos las clases Java que se encargan de realizar la función de controladores en nuestra aplicación web. Tienen la misma función que los servlets de Java EE pero se diferencian en las anotaciones utilizadas ya que estamos usando el framework Spring MVC. Para crear los controladores sólo será necesario incluir la anotación **@Controller** al principio de cada clase.

La atención de las peticiones web se realiza a través de los métodos del controlador. Cada método se encarga de gestionar las peticiones de una o más URLs asociadas a él, y se pueden crear tantos métodos como sean necesarios. Tener muchos métodos dentro de un controlador permite agrupar funcionalidad y da la



posibilidad de procesar una serie de URL relacionadas en un único controlador de forma sencilla (en un servlet había un único método para cubrir todas las peticiones).

Cada método de los controladores encargado de atender las peticiones web tendrá la anotación `@RequestMapping`, esta anotación tendrá como argumento una lista de cadenas de texto con todas las URLs que va a controlar. En nuestro caso existe un controlador para cada tabla pero si hay que destacar alguno debe ser el encargado de controlar la tabla Palabras. La ruta que utilizamos para realizar la traducción de texto a pictogramas tiene la siguiente estructura:

```
.../api/TraduceFrase/{frase}
```

Siendo frase la cadena de caracteres que queremos traducir.

Cuando un usuario introduce una frase en la aplicación Android, realizamos una petición a esta dirección URL y el controlador descrito se encarga de tratar esa petición realizando las funciones que hayan sido descritas. Es necesario explicar entonces el siguiente y último paquete.

6.3.3. *Com.ssiipicto.server.dao*

Este paquete contiene las clases que se encargan de realizar las consultas a la base de datos. Al haber aplicado el patrón DAO para independizar el acceso a datos de la tecnología utilizada para ello, debemos implementar las interfaces definidas en la parte de persistencia. De esta forma tenemos la clase *PalabrasDAOImplHibernate* que contiene toda la funcionalidad para resolver la llamada del cliente.



Capítulo 7:

AraTraductor

7. AraTraductor

Constituye el cliente del proyecto que será utilizado por los usuarios. Esta aplicación va destinada a las personas que quieren comunicarse con otras que tienen problemas de comunicación y no pueden comprender correctamente el lenguaje escrito. La aplicación está optimizada para ser utilizada en *tablets*, puesto que ofrece más facilidad y manejabilidad por el tamaño de las mismas.

Ha sido realizada con una interfaz intuitiva y de muy fácil uso. Comienza con una pantalla de presentación a modo de saludo.



ILUSTRACIÓN 22. INTERFAZ DE BIENVENIDA DE LA APLICACIÓN ARATRADUCTOR.

7.1. Funcionalidades implementadas

La funcionalidad principal es la de permitir al usuario introducir la frase que desea traducir. Para ello presenta una pantalla con un recuadro en el que introducir el texto deseado y un botón 'Traducir' para realizar la traducción. A la hora de introducir el texto a traducir se puede realizar mediante el teclado o por voz.



ILUSTRACIÓN 23. INTERFAZ PRINCIPAL DE LA APLICACIÓN ARATraductor.

Una vez que el usuario pulsa el botón 'Traducir' la aplicación realiza dicha traducción y se muestra en la siguiente pantalla, en la que se puede observar la frase introducida, la traducción realizada y cuatro botones. Como se puede apreciar en la Ilustración 24 con el ejemplo de la frase introducida "Hola amigos este es nuestro trabajo".

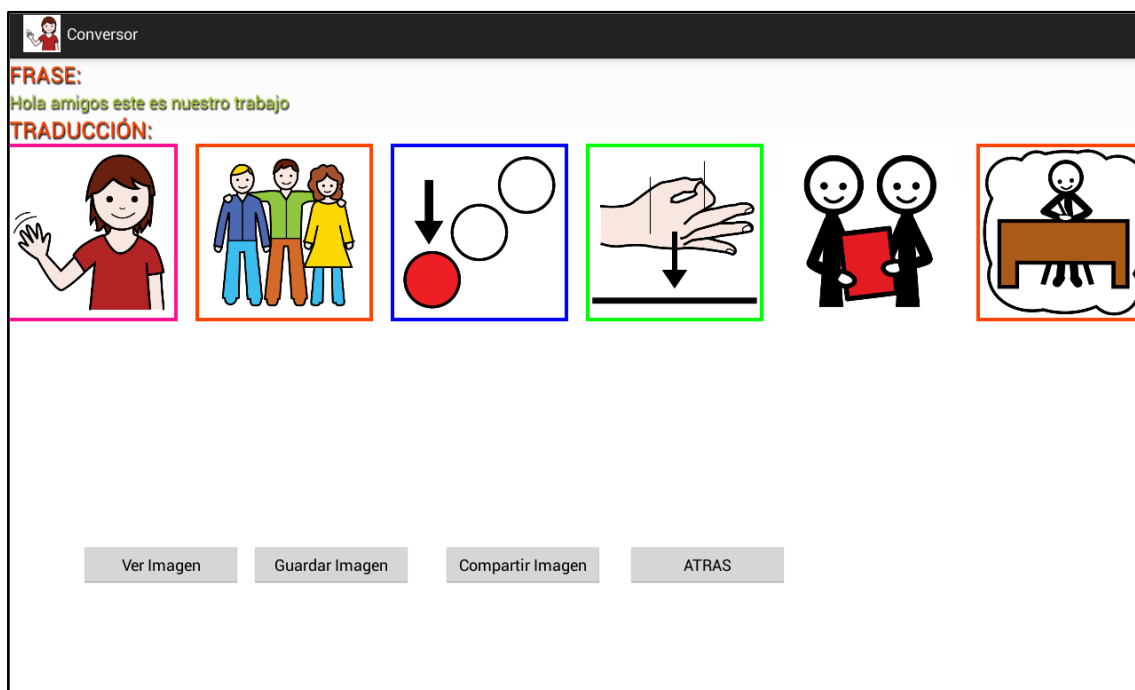


ILUSTRACIÓN 24. INTERFAZ DE RESULTADO DE LA TRADUCCIÓN.

La siguiente funcionalidad consiste en que al pulsar sobre una de las imágenes de la traducción, el sistema muestra una lista con todos los posibles pictogramas que se pueden utilizar para la palabra que referencia. Por ejemplo para la palabra "amigos" existen distintos pictogramas como se muestra en la Ilustración 25.

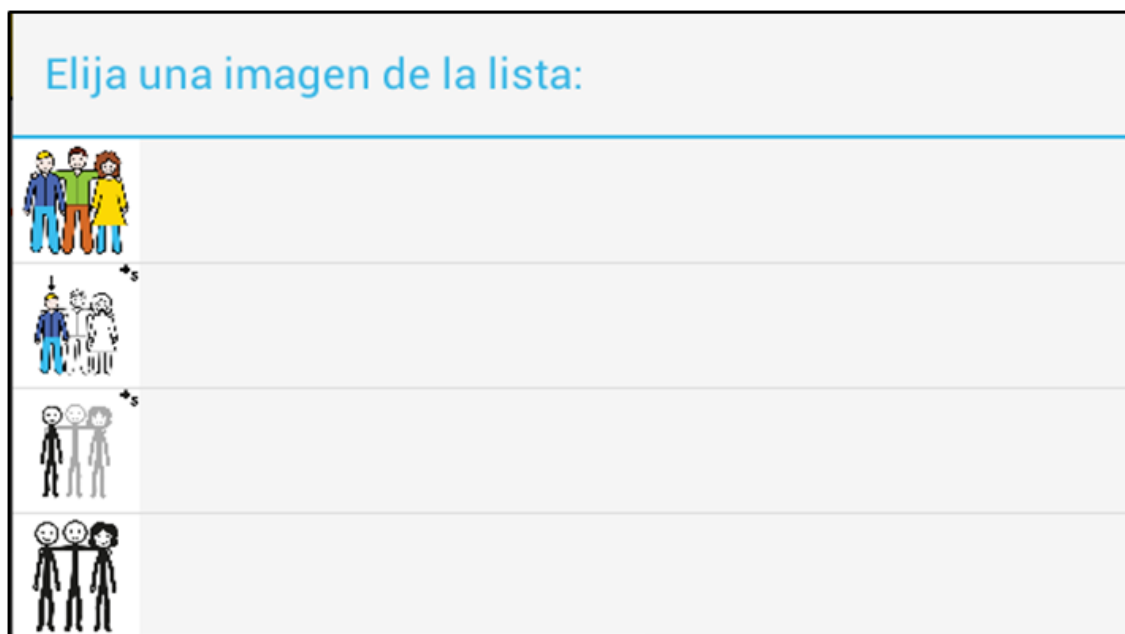


ILUSTRACIÓN 25. LISTADO DE LAS IMÁGENES A ELEGIR PARA UNA PALABRA.

Una vez que el usuario elige la imagen que desea de la lista, el sistema la cambia y muestra la imagen elegida en la traducción.

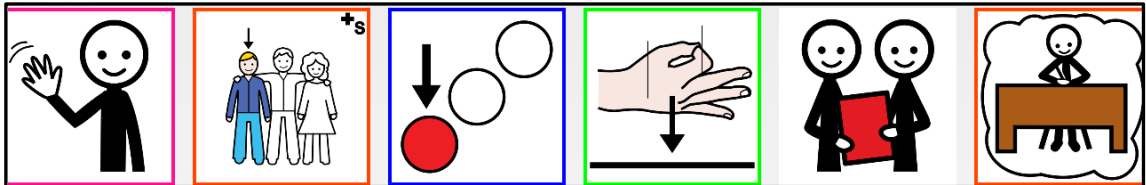


ILUSTRACIÓN 26. RESULTADO TRAS CAMBIAR ALGUNAS DE LAS IMÁGENES.

En cuanto a los botones que se presentan en la Ilustración 27, pasaremos a mostrar lo que realiza cada uno.



ILUSTRACIÓN 27. BOTONES DE LA INTERFAZ DE RESULTADO DE LA TRADUCCIÓN.

El botón 'ATRAS' sirve para regresar a la pantalla en la que el usuario puede escribir una frase para realizar la traducción.

El botón 'Ver Imagen' junta todos los pictogramas en una sola imagen .jpeg.

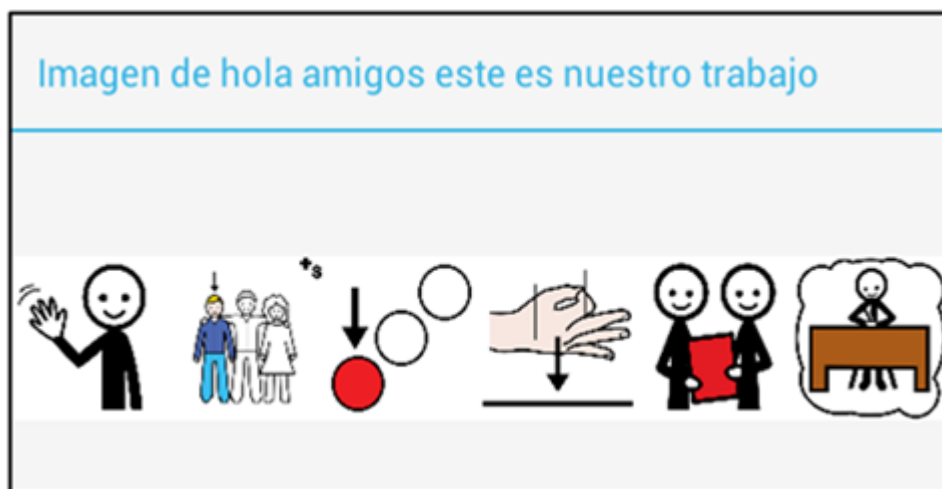


ILUSTRACIÓN 28. IMAGEN CREADA AL PULSAR EL BOTÓN 'VER IMAGEN'.

El botón 'Guardar Imagen' permite guardar la imagen en el terminal, de tal modo que en caso de que no exista, crea una carpeta con el nombre AraTraductor donde se guardaran las imágenes que se desee de la aplicación.

El botón 'Compartir Imagen' sirve para poder compartir la imagen por cualquiera de las aplicaciones que ofrece el sistema.

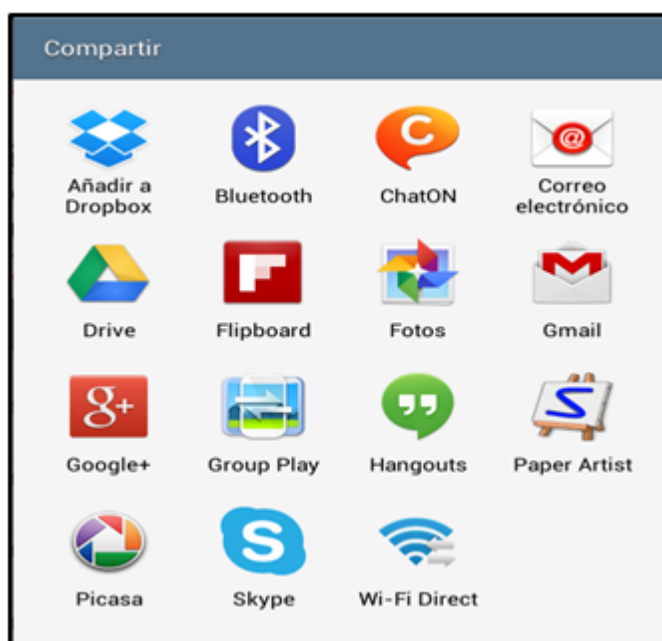


ILUSTRACIÓN 29. OPCIONES DE APLICACIONES CON LAS QUE COMPARTIR LA IMÁGEN.

En cuanto al tratamiento de palabras no encontradas en la base de datos, el sistema muestra una lista que incluye las palabras que no han sido encontradas junto con la posición en la que se encuentran. En la Ilustración 30 se puede observar que no ha encontrado las palabras 'aplicación' y 'aaa', y en su lugar muestra el pictograma de la interrogación de cierre.

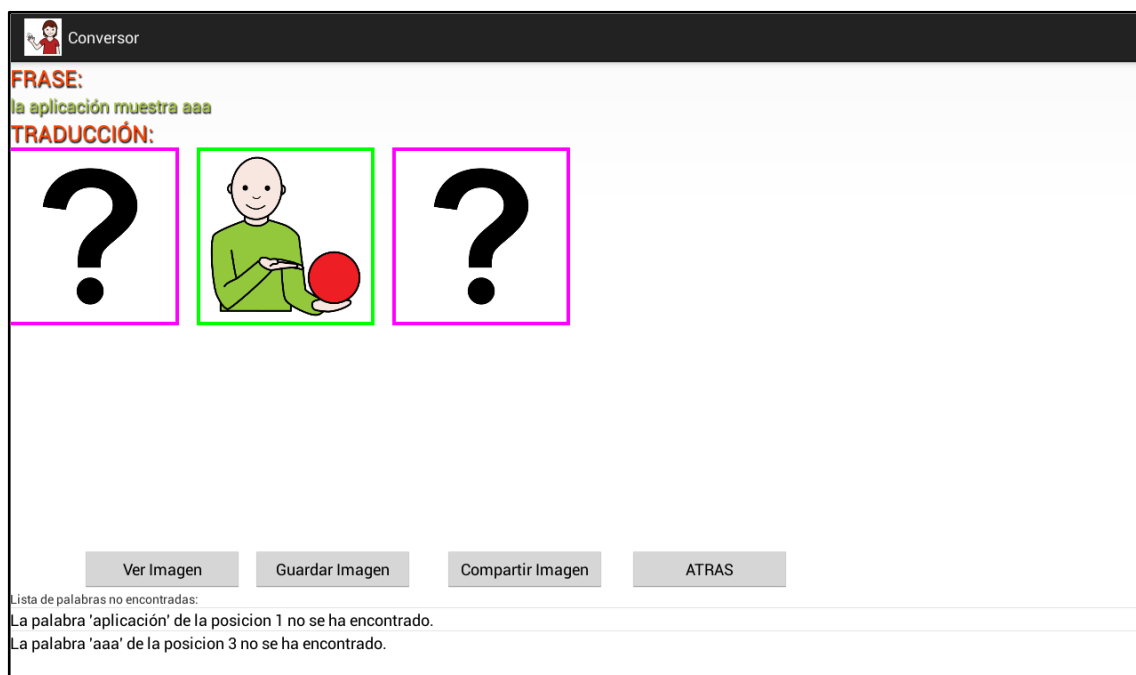


ILUSTRACIÓN 30. EJEMPLO DE TRADUCCIÓN CON PALABRAS NO ENCONTRADAS EN LA BASE DE DATOS.

7.2. Detalles de implementación

En este apartado se exponen los recursos más importantes utilizados para llevar a cabo el desarrollo de la aplicación.

Para la implementación se ha utilizado el entorno Eclipse Kepler, instalando el plug-in Android-SDK y el set de herramientas ADT (*Android Development Tools*). Se ha realizado para funcionar en versiones a partir de la versión Android 4.4, API 19.

Por motivos de seguridad, *Android* no permite la conexión a internet desde el hilo principal de ejecución, por lo que para realizar la llamada al servicio web es necesario crear una tarea asíncrona en la que se incluye dicha conexión. Se ha utilizado la librería *org.apache.http.** de modo que desde la aplicación enviamos la frase introducida por el usuario y una vez realizado el análisis completo de la frase, el servicio web nos devuelve un objeto de tipo JSON, que incluye la lista de palabras con sus pictogramas asociados.

Para tratar la información enviada por el servicio web hemos utilizado la librería *org.json.**, de modo que el servicio web encapsula dicha información en un objeto JSON y la aplicación lo recibe, realizando el desencapsulado del mismo y tratando la información recogida en su interior convenientemente.



Se ha decidido que si una palabra no es encontrada por el servicio web en la base de datos, la aplicación muestra un símbolo de interrogación en el lugar que le corresponde a dicha palabra y añade una lista indicando las palabras no encontradas junto con la posición en la que se encuentran en la traducción en pictogramas.



Capítulo 8: Evaluación

8. Evaluación

A modo de cierre del proyecto hemos realizado una serie de pruebas para comprobar la corrección de las traducciones proporcionadas por la herramienta creada. Para ello se ha considerado como documento de referencia o *gold standard* el documento de la Declaración de los Derechos Humanos traducida a pictogramas (Santos s.f.) y se ha comparado el resultado de la traducción ofrecida por otra herramienta anterior, ésta es Pictotraductor de la que se habló anteriormente en el capítulo 2, con el que se obtiene al utilizar nuestro sistema.

Pasamos a continuación a comentar las características de la Declaración de los Derechos Humanos traducida a pictogramas (que se ha tomado como *gold standard*), y de la aplicación Pictotraductor (que se ha tomado como *baseline*) para finalmente mostrar los casos más representativos de las pruebas realizadas.

8.1. *Gold standard y baseline*

Como se ha comentado anteriormente se ha utilizado como *gold standard* para la realización de las pruebas la Declaración de los Derechos Humanos traducida a pictogramas. El motivo de esta elección es que la traducción de este documento se ha realizado por seres humanos por lo que se considera que esta traducción representa el mayor grado de corrección posible que puede alcanzar cualquier herramienta informática destinada a ello. Se muestra, en la Ilustración 31, un ejemplo escogido para la realización de las pruebas que corresponde a la frase "Los estados partes protegerán y promoverán el derecho de las personas con discapacidad y sus familias a una vida adecuada incluyendo la alimentación, el vestido, y la vivienda sin discriminación" de la Declaración de los Derechos Humanos.

Asimismo se ha tomado como *baseline* la herramienta Pictotraductor que es un traductor de texto a pictogramas vía web, de la que se dio mayor detalle en el capítulo 2.



ILUSTRACIÓN 31. EJEMPLO DE FRASE DE LA DECLARACIÓN DE LOS DERECHOS HUMANOS.

8.2. Evaluación de los resultados

Como se ha comentado anteriormente se ha utilizado la aplicación Pictotraductor como *baseline* para evaluar nuestro sistema. Por lo tanto mostramos a continuación algunos de los casos probados más representativos, incluyendo para cada uno de ellos la frase original de la Declaración de los Derechos Humanos, el *baseline* y la traducción de nuestro sistema.

8.2.1. Prueba 1

La frase original de la Declaración de los Derechos Humanos es "¡Ánimo! Ahora puedes empezar a leer la convención en pictogramas", como se puede observar en la Ilustración 32. La traducción realizada por el Pictotraductor se puede ver en la Ilustración 33 y la realizada por AraTraductor en la Ilustración 34.



ILUSTRACIÓN 32. TRADUCCIÓN DE LA DECLARACIÓN DE LOS DERECHOS HUMANOS PARA LA PRUEBA 1.

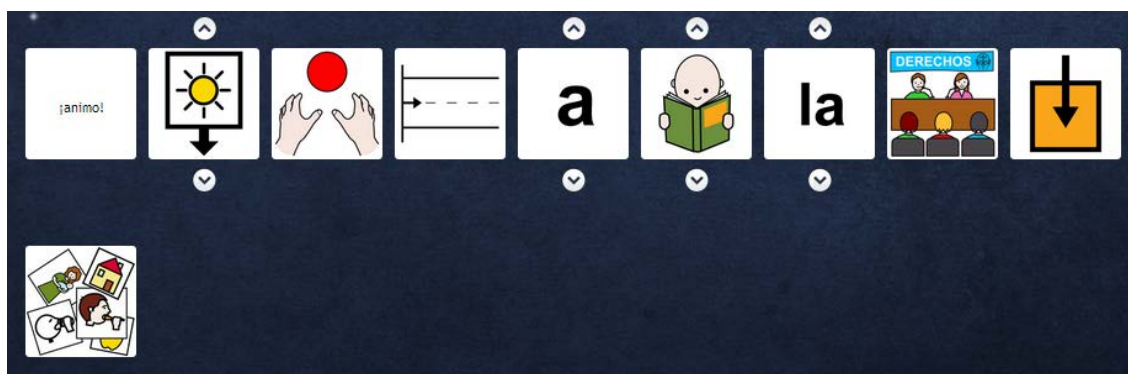


ILUSTRACIÓN 33. TRADUCCIÓN DE PICTOTRADUCTOR PARA LA PRUEBA 1.



ILUSTRACIÓN 34. TRADUCCIÓN DE ARATRADUCTOR PARA LA PRUEBA 1.



Como podemos observar en las ilustraciones de la prueba 1, la traducción del *gold standard* utiliza 9 pictogramas.

El *baseline* utiliza 10 pictogramas y si analizamos los errores cometidos, observamos que no utiliza pictogramas ni para la preposición 'a' ni para la palabra '¡Ánimo!'. Por último, muestra el artículo 'la', que no debería considerarse en la traducción.

AraTraductor realiza la traducción utilizando 11 pictogramas, el hecho de utilizar dos pictogramas más que el *gold standard* es por indicar el comienzo y fin de la expresión entre exclamaciones. El resto de pictogramas coinciden exactamente con los que aparecen en el *gold standard*.

8.2.2. Prueba 2

La frase original de la Declaración de los Derechos Humanos es "Los estados partes promoverán que la sociedad tome conciencia de las aportaciones y las capacidades de las personas con discapacidad.", como se puede observar en la Ilustración 35. La traducción realizada por el Pictotraductor se puede ver en la Ilustración 36 y la realizada por AraTraductor en la Ilustración 37.



ILUSTRACIÓN 35. TRADUCCIÓN DE LA DECLARACIÓN DE LOS DERECHOS HUMANOS PARA LA PRUEBA 2.

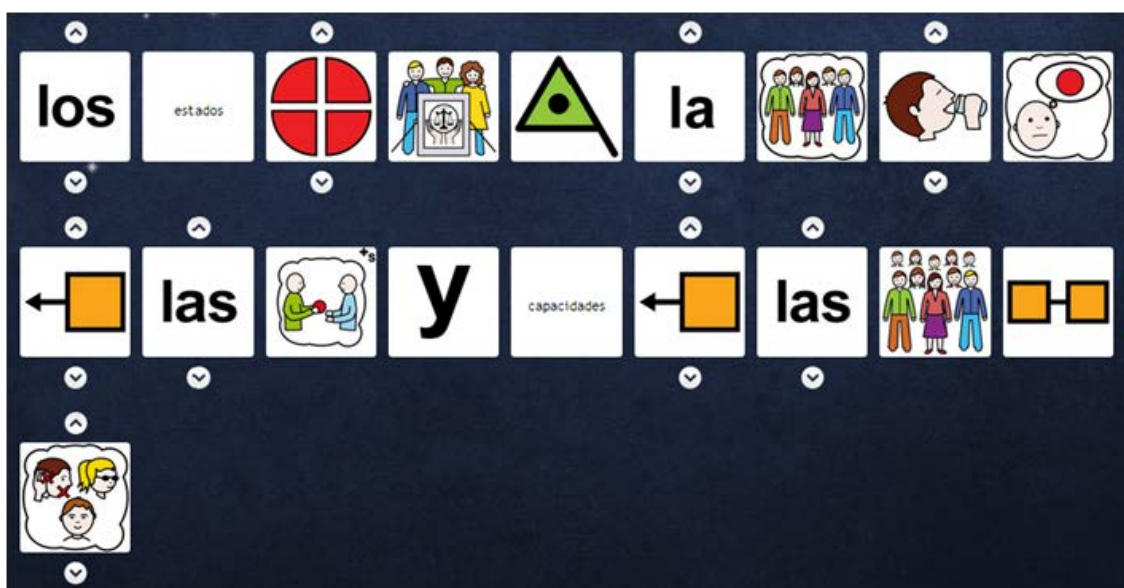


ILUSTRACIÓN 36. TRADUCCIÓN DE PICTOTRADUCTOR PARA LA PRUEBA 2.

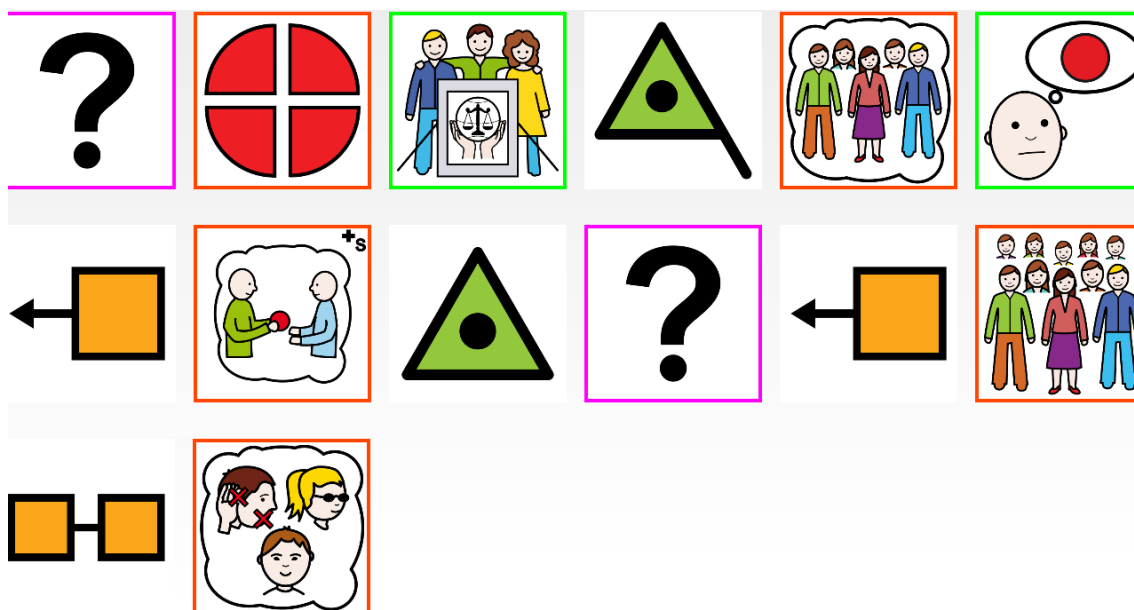


ILUSTRACIÓN 37. TRADUCCIÓN DE ARATRADUCTOR PARA LA PRUEBA 2.

Como podemos observar en las ilustraciones de la prueba 2, la traducción del *gold standard* utiliza 11 pictogramas.

El *baseline* utiliza 20 pictogramas y si analizamos los errores cometidos, observamos que se cometen cuatro errores por mostrar artículos que preceden a sustantivos, uno por no mostrar la imagen de la preposición 'y', por último, la expresión 'tomar conciencia' la separa en dos pictogramas en vez de utilizar el n-grama correspondiente.

AraTraductor realiza la traducción utilizando 14 pictogramas, el hecho de utilizar tres pictogramas más que el *gold standard* es por no contar con los pictogramas asociados a las expresiones 'Estados partes' y 'Personas con discapacidad' y 'capacidades' en la base de datos. Estos errores son comunes en el *baseline* y en AraTraductor.

8.2.3. Prueba 3

La frase original de la Declaración de los Derechos Humanos es "Los estados partes asegurarán el acceso a programas de vivienda pública y a apoyos en la jubilación", como se puede observar en la Ilustración 38. La traducción realizada por el Pictotraductor se puede ver en la Ilustración 39 y la realizada por AraTraductor en la Ilustración 40.



ILUSTRACIÓN 38. TRADUCCIÓN DE LA DECLARACIÓN DE LOS DERECHOS HUMANOS PARA LA PRUEBA 3.



ILUSTRACIÓN 39. TRADUCCIÓN DE PICTOTRADUCTOR PARA LA PRUEBA 3.

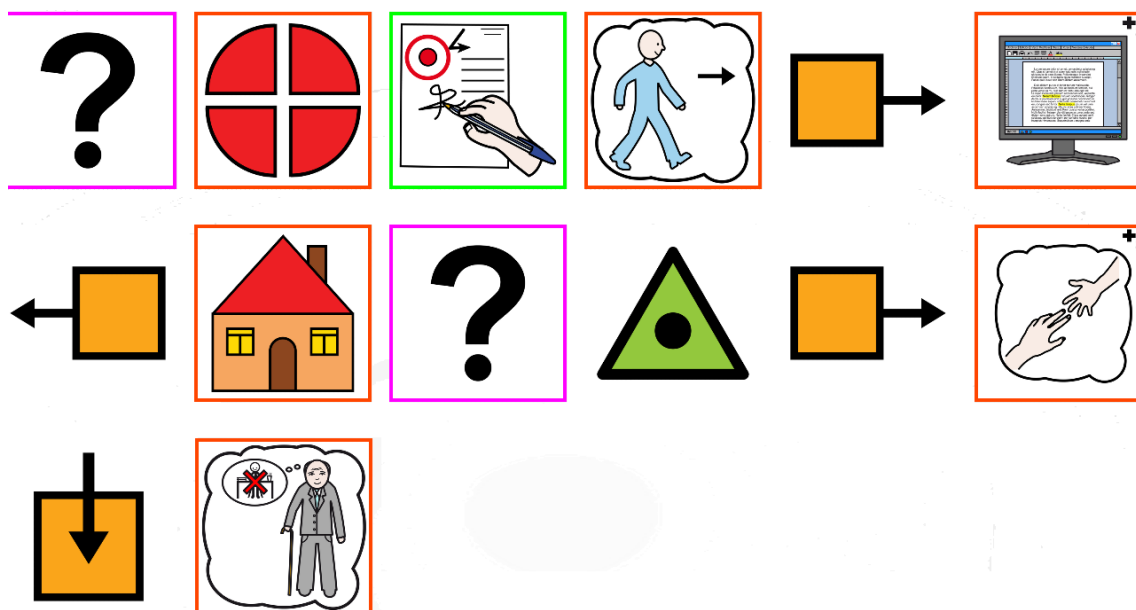


ILUSTRACIÓN 40. TRADUCCIÓN DE ARATRADUCTOR PARA LA PRUEBA 3.

Como podemos observar en las ilustraciones de la prueba 3, la traducción del *gold standard* utiliza 13 pictogramas.

El *baseline* utiliza 17 pictogramas y si analizamos los errores cometidos, observamos que se repite nuevamente el error de los artículos y aparece otro error en la traducción de la palabra 'asegurarán' puesto que muestra un pictograma diferente.

AraTraductor realiza la traducción utilizando 14 pictogramas, se utiliza un pictograma más que en el *gold standard* por no contar con el pictograma asociado a la expresión 'Estados partes', además no coincide el pictograma asociado a 'asegurarán'.

Además, ninguno de los dos sistemas cuenta con el pictograma de la palabra 'pública' en sus bases de datos.

8.2.4. Prueba 4

La frase original de la Declaración de los Derechos Humanos es " Los estados partes aseguraran el acceso a las personas con discapacidad incluidas las mujeres, las niñas y las personas mayores a los apoyos adecuados para la protección social contra la pobreza.", como se puede observar en la Ilustración 41. La traducción realizada por el Pictotraductor se puede ver en la Ilustración 42 y la realizada por AraTraductor en la Ilustración 43.

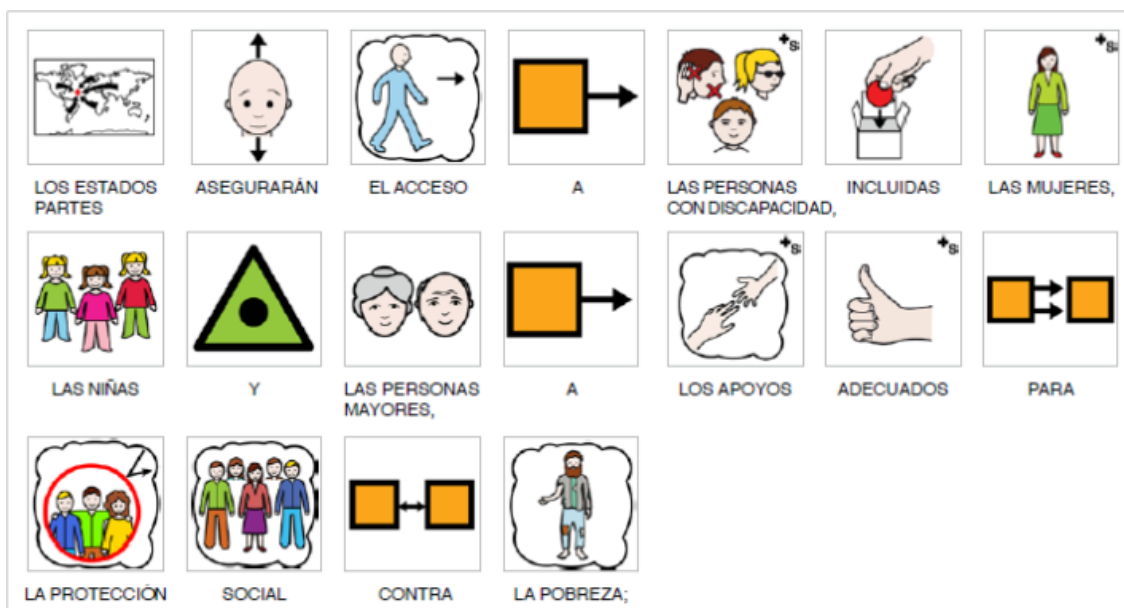


ILUSTRACIÓN 41. TRADUCCIÓN DE LA DECLARACIÓN DE LOS DERECHOS HUMANOS PARA LA PRUEBA 4.



ILUSTRACIÓN 42. TRADUCCIÓN DE PICTOTRADUCTOR PARA LA PRUEBA 4.

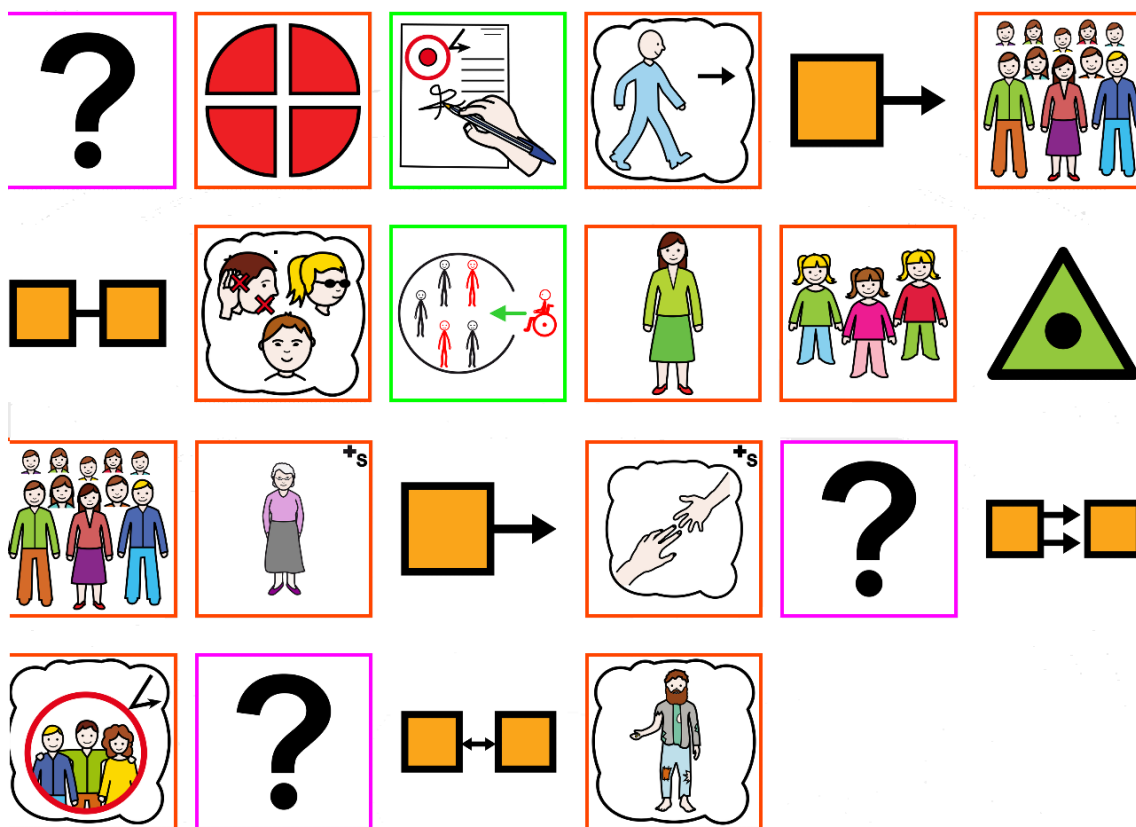


ILUSTRACIÓN 43. TRADUCCIÓN DE ARATRADUCTOR PARA LA PRUEBA 4.

Como podemos observar en las ilustraciones de la prueba 4, la traducción del *gold standard* utiliza 18 pictogramas.

El *baseline* utiliza 31 pictogramas y si analizamos los errores cometidos, observamos que se repiten los mismos errores que en las pruebas anteriores. Introduce nueve artículos innecesarios y no dispone de pictograma para cuatro palabras: 'incluidas', 'mujeres', 'adecuados' y 'social'.

AraTraductor realiza la traducción utilizando 22 pictogramas, evitando todos los errores por artículos y no disponiendo de pictograma para dos palabras: 'adecuados' y 'social'.

El hecho de encontrar dos palabras más que el *baseline* se debe a la implementación de un algoritmo mejor para realizar la traducción, ya que en caso de no encontrar la palabra original pedida por el usuario se busca su lema.

8.2.5. Prueba 5

La frase original de la Declaración de los Derechos Humanos es "Los países que quieran pueden firmar la convención en Naciones Unidas desde Marzo del año 2007.", como se puede observar en la Ilustración 44. La traducción realizada por el Pictotraductor se puede ver en la Ilustración 45 y la realizada por AraTraductor en la Ilustración 46.



ILUSTRACIÓN 44. TRADUCCIÓN DE LA DECLARACIÓN DE LOS DERECHOS HUMANOS PARA LA PRUEBA 5.

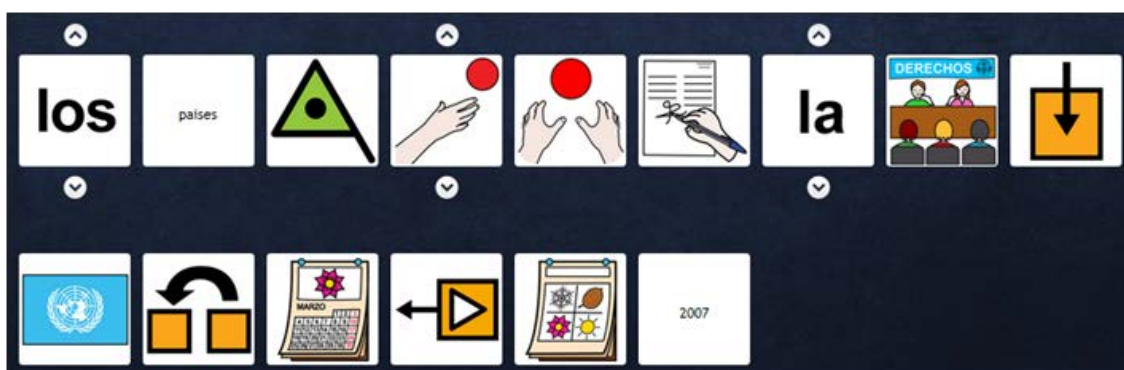


ILUSTRACIÓN 45. TRADUCCIÓN DE LA PICTOTRADOR PARA LA PRUEBA 5.

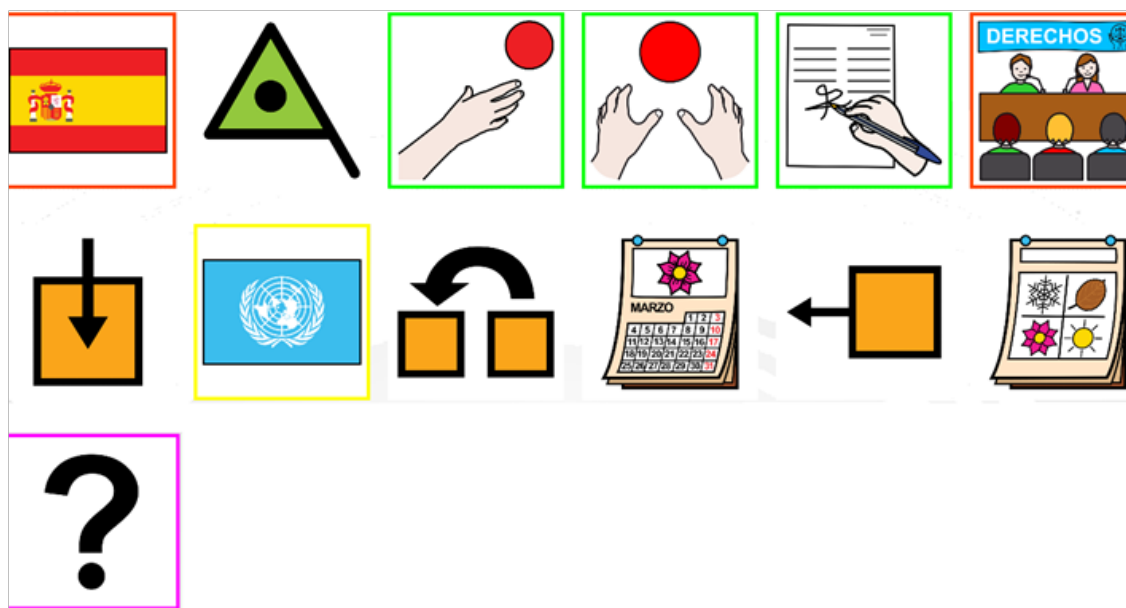


ILUSTRACIÓN 46. TRADUCCIÓN DE LA AraTraductor PARA LA PRUEBA 5.

Como podemos observar en las ilustraciones de la prueba 5, la traducción del *gold standard* utiliza 13 pictogramas.

El *baseline* utiliza 15 pictogramas repitiendo los errores explicados en las pruebas anteriores mientras que AraTraductor se sirve del mismo número de pictogramas para realizar su traducción, cometiendo errores de no disponer del mismo pictograma que el *gold standard* para dos palabras y no encontrando un pictograma para '2007'.

8.3. Discusión de los resultados obtenidos

Como se puede apreciar en las pruebas mostradas, nuestro sistema ofrece traducciones notablemente más cercanas a las que aparecen en la Declaración de los Derechos Humanos respecto a las que proporciona la aplicación Pictotraductor, ya que esta última no realiza un análisis profundo de la frase, ni elimina determinantes (innecesarios para el significado global del texto) y, en ocasiones, cuando la frase es excesivamente larga como en el caso de la prueba 4, ni siquiera proporciona una traducción aceptable.

Asimismo, también es cierto que las traducciones ofrecidas por nuestro sistema presentan ligeras variaciones o errores respecto a la que aparece en el *gold*

standard. Pasamos por tanto a comentar los tipos de errores y a explicar la aparición de los mismos en nuestro sistema.

Fundamentalmente los errores aparecidos se deben a dos causas: errores por carencia del pictograma en la base de datos y errores por análisis incorrecto de Maltparser.

8.3.1. Errores por carencia del pictograma en la base de datos

Como ya se ha comentado anteriormente, la base de datos utilizada en el proyecto se ha desarrollado a partir de la que nos proporcionó la asociación ARASAAC. Sin que se nos hubiese facilitado este material habría sido imposible la realización del proyecto. Sin embargo, es cierto que esta base de datos carece de algunos de los pictogramas con los que sí cuenta la traducción de la Declaración de los Derechos Humanos, tales como el pictograma asociado a "Los estados partes" (Ilustración 47).



ILUSTRACIÓN 47. PICTOGRAMA DE "LOS ESTADOS PARTES" DE LA DECLARACIÓN DE LOS DERECHOS HUMANOS.

Por tanto, es evidente que sin contar con este material en la base de datos resulta imposible hacer una traducción correcta de la frase que corresponda.

A este tipo de error corresponde, como se ha comentado anteriormente, la aparición de todos los pictogramas con el símbolo de interrogación que se observan en la traducciones de nuestro sistema. Sin embargo también es cierto que nuestra aplicación mitiga en parte este error al informar al usuario de que efectivamente la base de datos no cuenta con un pictograma para la palabra o palabras en cuestión, indicando además su posición dentro del texto.

La solución total de este error consistiría simplemente en una actualización de la base de datos con los pictogramas correspondientes, lo que no significa más que

una única inserción en las tablas creadas que se comentaron en el capítulo correspondiente.

8.3.2. Errores por análisis incorrecto de Maltparser

El otro tipo de error que aparece en las traducciones ofrecidas por nuestro sistema se deben simplemente a errores en el análisis que realiza del texto el servicio web Maltparser.

Desgraciadamente el sistema Maltparser no es infalible, por lo que para algunos casos no proporciona un análisis correcto del texto lo que desemboca inevitablemente en una traducción poco acertada de nuestro sistema. Sin embargo este error sería nuevamente subsanable si se entrenara de manera correcta el sistema Maltparser para nuestros propósitos específicos.

Como se aprecia en la Ilustración 48, al pasarle como argumento la frase "Yo como patatas fritas", nos devuelve como resultado que la palabra "como" es un complemento y que el verbo es "fritas" lo que es evidentemente incorrecto y provocaría una mala traducción por parte de nuestro sistema.

| | | | | | | | | |
|---|---------|--------|---|----------|---|------|------|---|
| 1 | Yo | yo | p | PP1CSN00 | — | 0 | ROOT | — |
| 2 | como | como | c | CS | 1 | MOD | — | — |
| 3 | patatas | patata | n | NCFP000 | 2 | COMP | — | — |
| 4 | fritas | freir | v | VMP00PF | 3 | MOD | — | — |

ILUSTRACIÓN 48. EJEMPLO DE ANÁLISIS INCORRECTO DE MALTPARSER.

8.4. Indicadores y métricas de los resultados

En este apartado vamos a exponer y analizar los resultados obtenidos por AraTraductor y por PictoTraductor en las veintiuna pruebas realizadas, comparando siempre con el *gold standard* que hemos tomado, que recordemos, es la Declaración de los Derechos Humanos. Todas las pruebas pueden verse en el apéndice A.

Cada prueba consiste en realizar la traducción de una frase sacada de modo aleatorio de la Declaración de los Derechos Humanos y comprobar el grado de similitud entre ella y las traducciones de los sistemas. Para ello vamos a utilizar las métricas *Precision* y *Recall*, siendo la precisión (*Precision*) la fracción de pictogramas recuperados que son relevantes y la exhaustividad (*Recall*) es la fracción de

pictogramas relevantes que son recuperados. Para el cálculo de estas métricas clasificamos los pictogramas en función de los siguientes indicadores:

- *True positive* (Tp): Pictogramas que están en la Declaración de los Derechos Humanos y también están en la traducción realizada por el sistema evaluado.
- *False negative* (Fn): Pictogramas que están en la Declaración de los Derechos Humanos y no están en la traducción realizada por el sistema evaluado.
- *True negative* (Tn): Pictogramas que no están en la Declaración de los Derechos Humanos y no están en la traducción realizada por el sistema evaluado.
- *False positive* (Fp): Pictogramas que no están en la Declaración de los Derechos Humanos y están en la traducción realizada por el sistema evaluado.

Las fórmulas aplicadas para obtener los resultados son las siguientes:

$$Precision = \frac{Tp}{Tp+Fp} \qquad Recall = \frac{Tp}{Tp+Fn}$$

Además también contabilizamos el número total de pictogramas utilizados por las aplicaciones en comparación con el *gold standard*.

En la tabla de la Ilustración 49 se pueden ver los resultados. Las columnas “B” corresponden al *baseline* (PictoTraductor), y las columnas “A” a nuestro sistema (AraTraductor).



| Prueba | Número pictogramas | | | Tp | | Fp | | Tn | | Fn | | Precision | | Recall | |
|--------|--------------------|-----|-----|-----|-----|-----|----|----|---|----|----|-----------|---------|---------|---------|
| | Gold Standard | B | A | B | A | B | A | B | A | B | A | B | A | B | A |
| 1 | 9 | 10 | 11 | 7 | 9 | 3 | 2 | 0 | 0 | 2 | 0 | 70,00% | 81,82% | 77,78% | 100,00% |
| 2 | 11 | 18 | 13 | 10 | 11 | 8 | 2 | 0 | 0 | 1 | 0 | 55,56% | 84,62% | 90,91% | 100,00% |
| 3 | 15 | 20 | 17 | 13 | 13 | 7 | 4 | 0 | 0 | 4 | 1 | 65,00% | 76,47% | 76,47% | 92,86% |
| 4 | 9 | 15 | 13 | 6 | 8 | 9 | 5 | 0 | 0 | 3 | 1 | 40,00% | 61,54% | 66,67% | 88,89% |
| 5 | 11 | 20 | 14 | 10 | 11 | 9 | 3 | 0 | 0 | 1 | 0 | 52,63% | 78,57% | 90,91% | 100,00% |
| 6 | 6 | 8 | 6 | 5 | 6 | 3 | 0 | 0 | 0 | 1 | 0 | 62,50% | 100,00% | 83,33% | 100,00% |
| 7 | 21 | 30 | 24 | 13 | 19 | 17 | 5 | 0 | 0 | 8 | 2 | 43,33% | 79,17% | 61,90% | 90,48% |
| 8 | 16 | 23 | 19 | 11 | 12 | 12 | 6 | 0 | 0 | 5 | 3 | 47,83% | 66,67% | 68,75% | 80,00% |
| 9 | 11 | 20 | 15 | 11 | 10 | 9 | 5 | 0 | 0 | 0 | 1 | 55,00% | 66,67% | 100,00% | 90,91% |
| 10 | 13 | 17 | 14 | 8 | 14 | 9 | 3 | 0 | 0 | 5 | 2 | 47,06% | 82,35% | 61,54% | 87,50% |
| 11 | 6 | 7 | 6 | 4 | 5 | 3 | 1 | 0 | 0 | 2 | 1 | 57,14% | 83,33% | 66,67% | 83,33% |
| 12 | 19 | 26 | 24 | 16 | 18 | 10 | 6 | 0 | 0 | 3 | 1 | 61,54% | 75,00% | 84,21% | 94,74% |
| 13 | 11 | 18 | 13 | 10 | 11 | 8 | 2 | 0 | 0 | 1 | 0 | 55,56% | 84,62% | 90,91% | 100,00% |
| 14 | 18 | 24 | 21 | 13 | 15 | 11 | 6 | 0 | 0 | 5 | 3 | 54,17% | 71,43% | 72,22% | 83,33% |
| 15 | 18 | 31 | 22 | 13 | 15 | 18 | 7 | 0 | 0 | 5 | 3 | 41,94% | 68,18% | 72,22% | 83,33% |
| 16 | 8 | 8 | 8 | 7 | 8 | 1 | 0 | 0 | 0 | 1 | 0 | 87,50% | 100,00% | 87,50% | 100,00% |
| 17 | 17 | 23 | 20 | 12 | 16 | 11 | 4 | 0 | 0 | 5 | 1 | 52,17% | 80,00% | 70,59% | 94,12% |
| 18 | 16 | 22 | 19 | 12 | 13 | 10 | 6 | 0 | 0 | 4 | 3 | 54,55% | 68,42% | 75,00% | 81,25% |
| 19 | 13 | 17 | 14 | 8 | 11 | 9 | 3 | 0 | 0 | 5 | 2 | 47,06% | 78,57% | 61,54% | 84,62% |
| 20 | 13 | 15 | 13 | 13 | 11 | 2 | 2 | 0 | 0 | 0 | 2 | 86,67% | 84,62% | 100,00% | 84,62% |
| 21 | 21 | 30 | 24 | 14 | 19 | 16 | 5 | 0 | 0 | 7 | 2 | 46,67% | 79,17% | 66,67% | 90,48% |
| TOTAL | 282 | 402 | 330 | 216 | 255 | 185 | 77 | 0 | 0 | 68 | 28 | 56,37% | 78,63% | 77,42% | 90,97% |

ILUSTRACIÓN 49. RESULTADOS DE LAS PRUEBAS.

El número de pictogramas mostrados por el *gold standard* es 282, el *baseline* utiliza un total de 402 pictogramas para realizar la traducción de las mismas frases y AraTraductor utiliza 330 (Ilustración 50). Cabe mencionar que aproximadamente 35 pictogramas son utilizados de más por la falta de pictogramas como “los estados partes” y “las personas con discapacidad” por lo que los resultados tanto del *baseline* como de AraTraductor mejorarían notablemente con la inclusión de estos pictogramas en la base de datos.

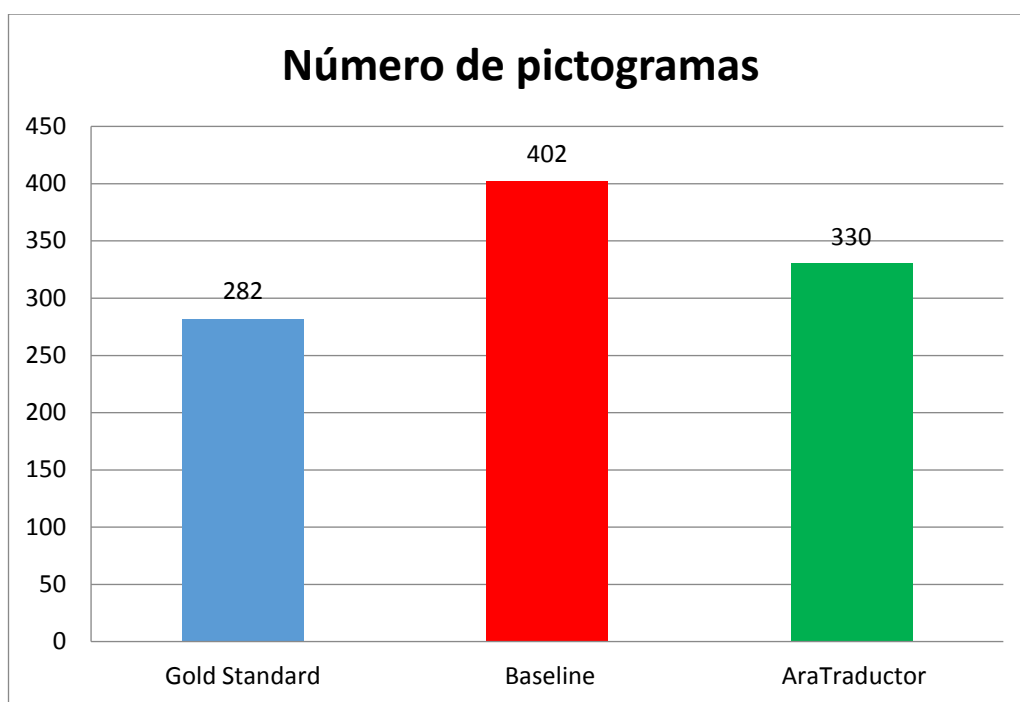


ILUSTRACIÓN 50. NÚMERO DE PICTOGRAMAS.

La precisión del *baseline* es exactamente 56,37% y la de AraTraductor es 78,63% (Ilustración 51).

Esta medida representa el porcentaje de pictogramas que coinciden exactamente con la Declaración de los Derechos Humanos.

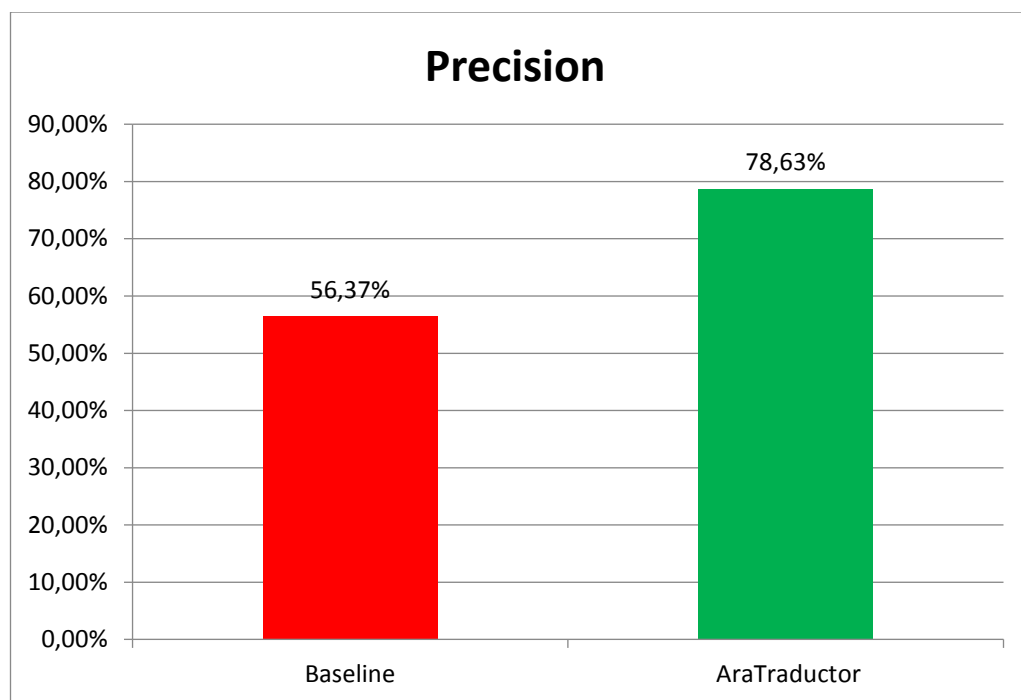


ILUSTRACIÓN 511. PRECISION.

La exhaustividad del *baseline* es exactamente 77,42%, y la de AraTraductor es de 90,97% (Ilustración 52).

Esta medida representa el número de pictogramas de la Declaración de los Derechos Humanos que son correctamente traducidos por los sistemas.

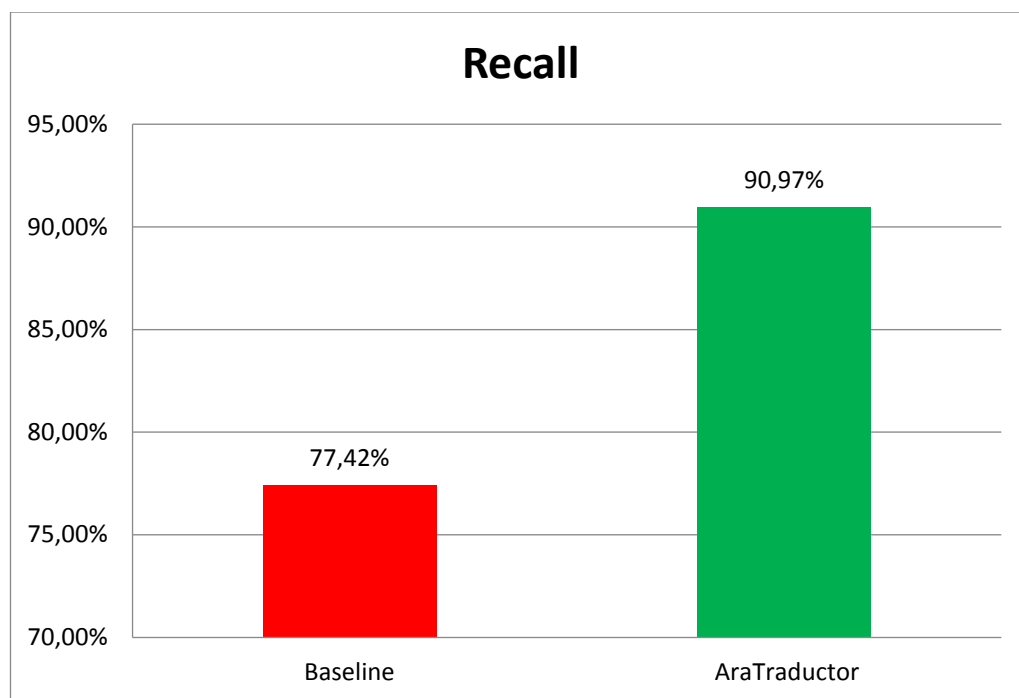


ILUSTRACIÓN 52. RECALL.

En definitiva, podemos concluir que hemos mejorado notablemente el funcionamiento de PictoTraductor creando una herramienta que realiza una traducción casi tan precisa como la realizada por una persona, y sólo mejorable, como hemos mencionado anteriormente, con la inclusión de nuevos pictogramas en la base de datos y mejorando el análisis sintáctico que realiza *MaltParser*.



Capítulo 9: Conclusiones y trabajo futuro

9. Conclusiones y trabajo futuro

9.1. Discusión

Pasamos en este apartado a comentar algunos aspectos que a nuestro parecer son merecedores de una discusión especial, al habernos resultado especialmente difíciles o simplemente más curiosos o interesantes.

9.1.1. Servicios web pesados vs servicios web ligeros

Como se ha comentado en varias ocasiones a lo largo de este documento, el proyecto ha sufrido durante su desarrollo un cambio total de su arquitectura motivado únicamente por el hecho de que al haber realizado en un primer momento la misma como un servicio web pesado, nos encontramos posteriormente con infinidad de dificultades, tanto durante el desarrollo de la parte del servidor, como para lograr un uso correcto de las funcionalidades por parte del cliente. Esto ha sido debido a la estructura de datos que utiliza para encapsular de forma automática el propio servicio web para comunicarse con el cliente, incrementando notablemente la dificultad del uso de la misma.

Finalmente y tras investigar nuevas posibilidades nos decantamos por la realización de un servicio web ligero considerablemente más sencillo de desarrollar y a la vez de comunicar con el cliente al requerir únicamente del uso de la clase JSON.

Podríamos decir que hemos tenido este problema por duplicado, ya que para comunicarnos con el servicio web de Maltparser que utilizamos para realizar el análisis del texto a traducir, nos encontramos con que al estar desarrollado como servicio web pesado resultaba incompatible realizar una llamada directamente desde nuestro servicio web, lo que nos obligó a buscar una solución alternativa consistente en crear un archivo .jar que realizase la llamada y ejecutar dicho archivo desde nuestro servicio web.

9.1.2. Errores en el análisis por parte de Maltparser

Como se ha comentado en el capítulo anterior, el sistema Maltparser del que hacemos uso para realizar el análisis del texto no es infalible y por tanto para algunos casos no funciona tal y como se espera. Por este motivo, el sistema proporciona, para

un reducido grupo de casos, traducciones incorrectas. Nos sentimos realmente impotentes ante la aparición de este tipo de error que no está en nuestra mano solucionar, pero sin embargo hay que reconocer que nos decantamos por el uso de Maltparser ya que es notablemente mejor al resto de opciones estudiadas tanto por corrección como por facilidad de uso.

9.1.3. Problemas en el desarrollo del cliente Android

Durante el desarrollo del cliente Android hemos encontrado un problema a la hora de mostrar la imagen compuesta de los pictogramas que forman la traducción o al compartir la misma. Este problema consiste en que si la frase consta de más de ocho pictogramas la imagen no se muestra debido al redimensionamiento de la misma. Este problema queda incluido en el apartado de corrección de errores de AraTraductor del apartado 9.3.1.

9.2. Conclusiones

Por todo lo comentado a lo largo del documento y en vista de la evaluación realizada, concluimos que se cumple con los objetivos inicialmente planteados en el apartado 1.2.

Asimismo consideramos que la traducción que logramos con nuestro sistema mejora la proporcionada por otros sistemas de la misma índole contribuyendo así a reducir la exclusión social de personas con dificultades en la comunicación.

9.3. Trabajo futuro

Durante el proceso de desarrollo del proyecto han surgido nuevas necesidades y mejoras que no han podido ser llevadas a cabo pero incrementarían el valor y la funcionalidad del proyecto. Vamos a dividir las posibles mejoras en tres módulos diferentes: aplicación Android, servicios web y recursos externos.

9.3.1. Trabajo futuro en AraTraductor

Las posibles ampliaciones que se podrían realizar son las siguientes:

- Posibilidad de creación de pictogramas a través de la cámara realizando fotografías e incluyéndolas en una base de datos propia de cada dispositivo.
- En la base de datos propia del dispositivo se buscarían las imágenes antes de realizar la búsqueda en la base de datos general ubicada en el servidor.
- Mejoras en la usabilidad y en la interfaz.
- Posibilidad de añadir nueva funcionalidad como juegos rápidos relacionados con los pictogramas, como por ejemplo, un juego de adivinar el significado de los pictogramas que se muestren, un juego de emparejar pictogramas de una misma palabra, etc.
- Adaptación de la aplicación para su uso en dispositivos móviles puesto que actualmente se encuentra optimizada para *tablets*.
- Corrección de errores.

9.3.2. Trabajo futuro en la base de datos

Las posibles ampliaciones que se podrían realizar son las siguientes:

- Añadir nuevos pictogramas y nuevas palabras ya que la principal limitación de nuestra aplicación es la falta de imágenes para muchas palabras de uso común.
- Permitir a personas relacionadas con ARASAAC poder ampliar de forma automática la base de datos.

9.3.3. Trabajo futuro en el traductor de texto a pictogramas

Las posibles ampliaciones que se podrían realizar son las siguientes:

- Mejorar la traducción a pictogramas, buscando sinónimos de palabras que no tengan un pictograma asociado hasta encontrar una imagen.
- Trabajar con el tiempo verbal para poder indicar si fue en tiempo pasado, presente o futuro. Para llevar a cabo esto el servicio web de Maltparser debería evolucionar y ser más eficiente y seguro.



- Mejorar el tiempo de respuesta del servicio web, incluyendo la llamada al servicio web de MaltParser.



Capítulo 10: Apéndices

10. Apéndices

Se incluye a continuación un apéndice mostrando la totalidad de las pruebas realizadas.

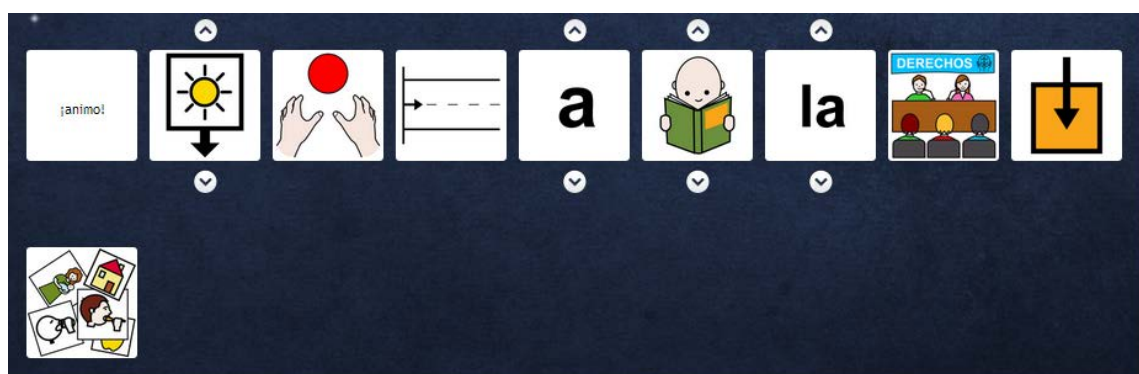
10.1. Apéndice A

FRASE 1: "¡Ánimo! Ahora puedes empezar a leer la convención en pictogramas."

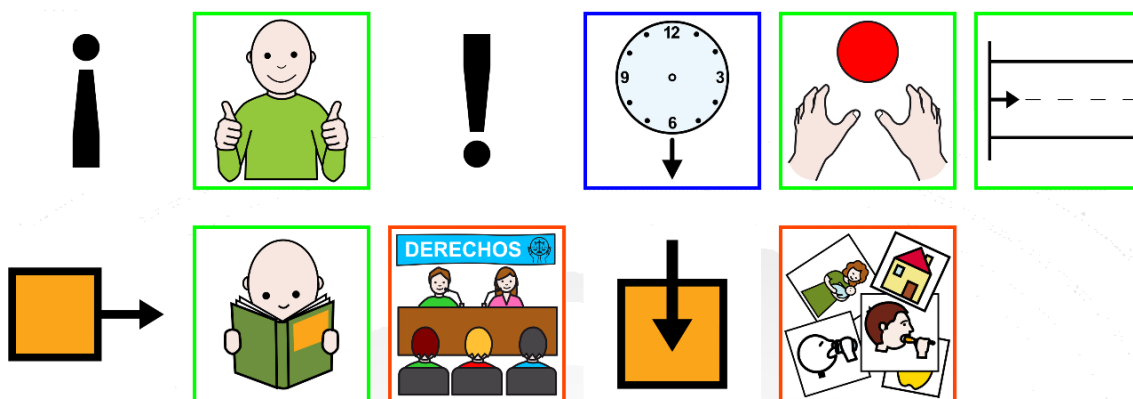
Declaración de los Derechos Humanos:



PictoTraductor:

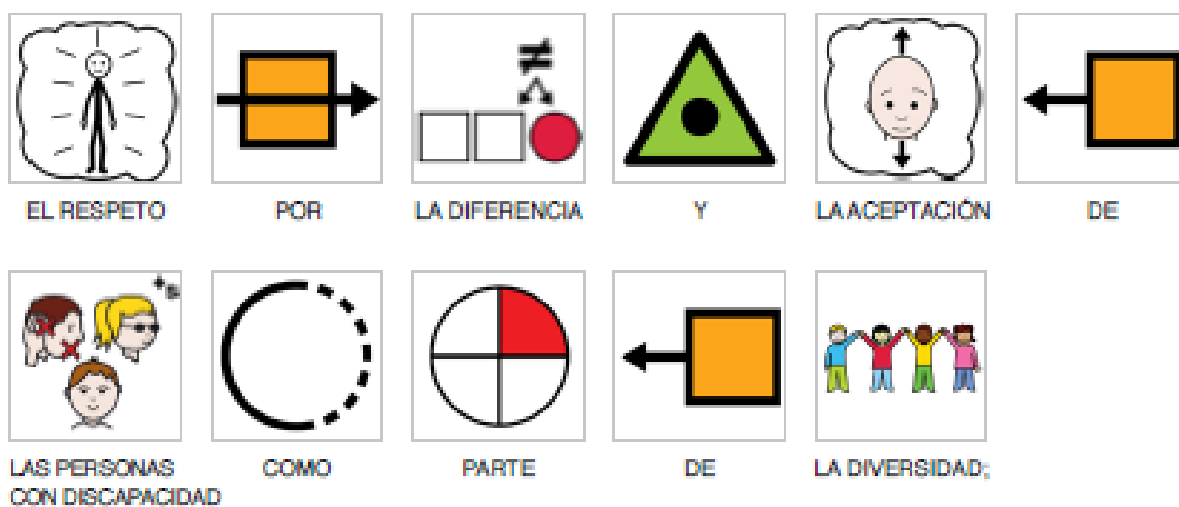


AraTraductor:



FRASE 2: "El respeto por la diferencia y la aceptación de las personas con discapacidad como parte de la diversidad."

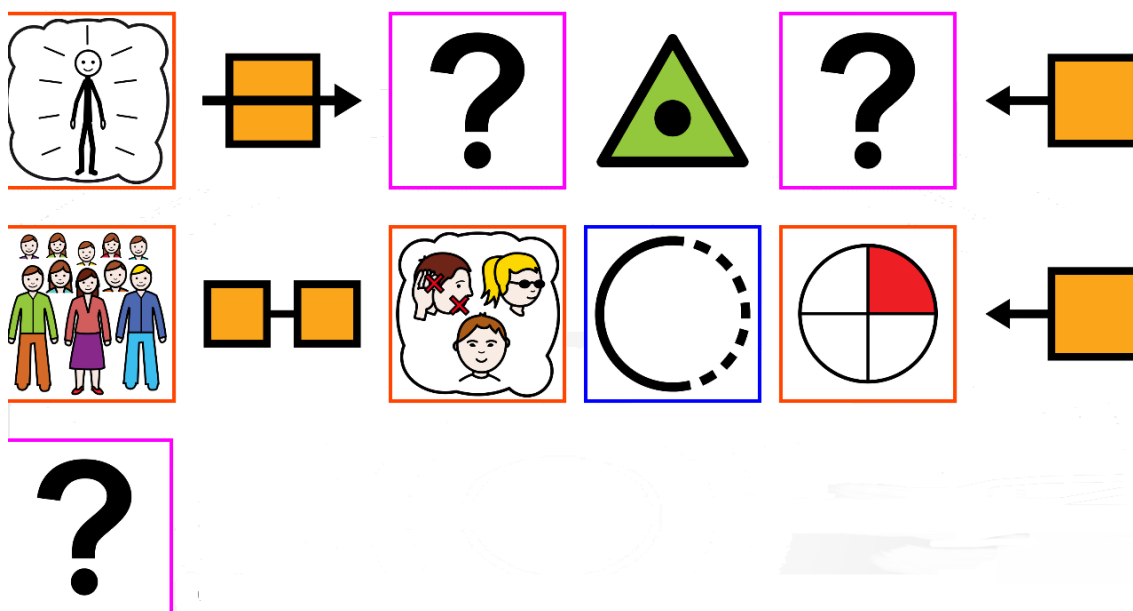
Declaración de los Derechos Humanos:



PictoTraductor:



AraTraductor:

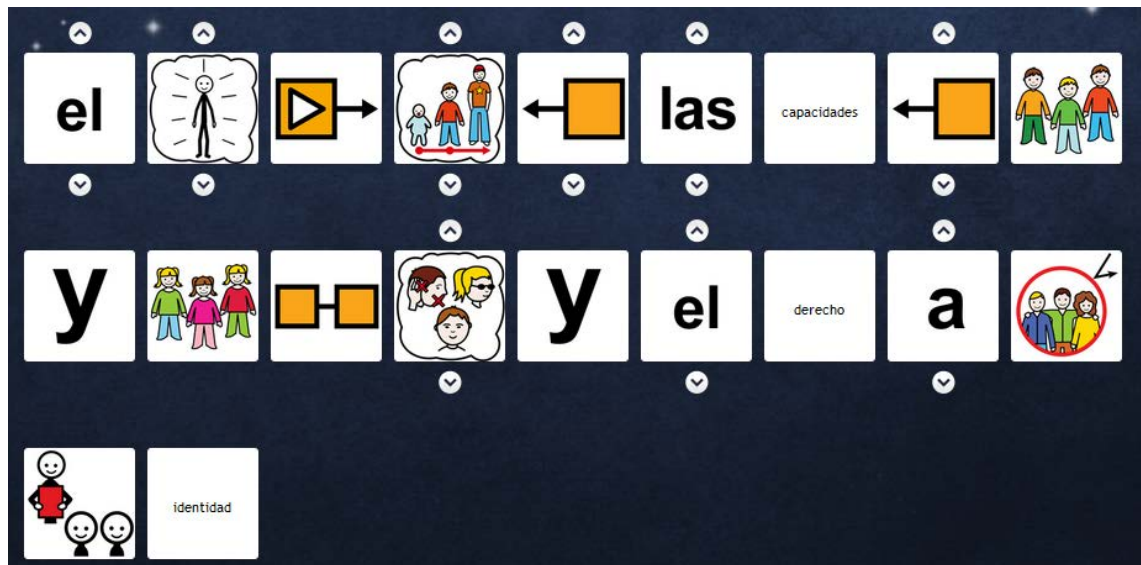


FRASE 3: "El respeto al desarrollo de las capacidades de niños y niñas con discapacidad y el derecho a proteger su identidad."

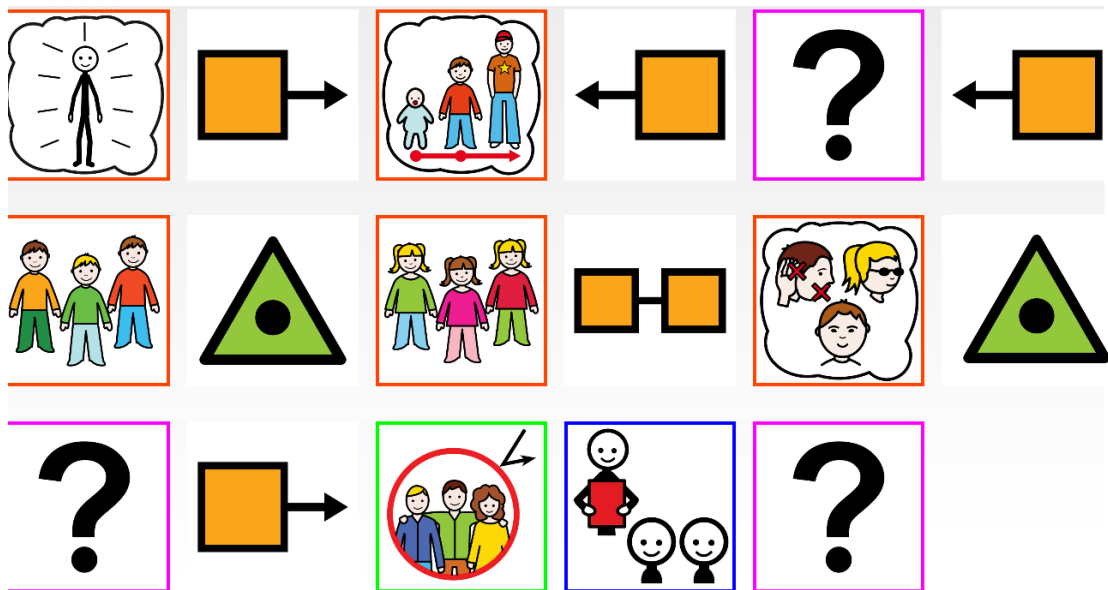
Declaración de los Derechos Humanos:



PictoTraductor:



AraTraductor:

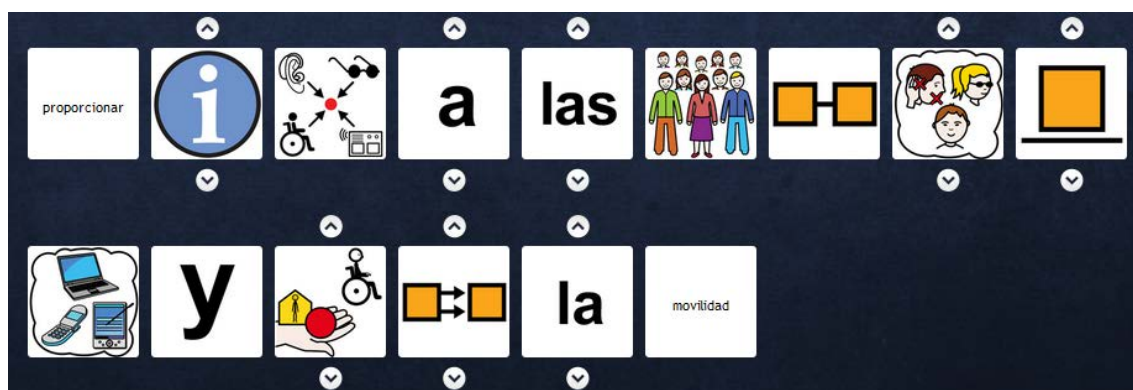


FRASE 4: "Proporcionar información accesible a las personas con discapacidad sobre tecnología y ayudas para la movilidad."

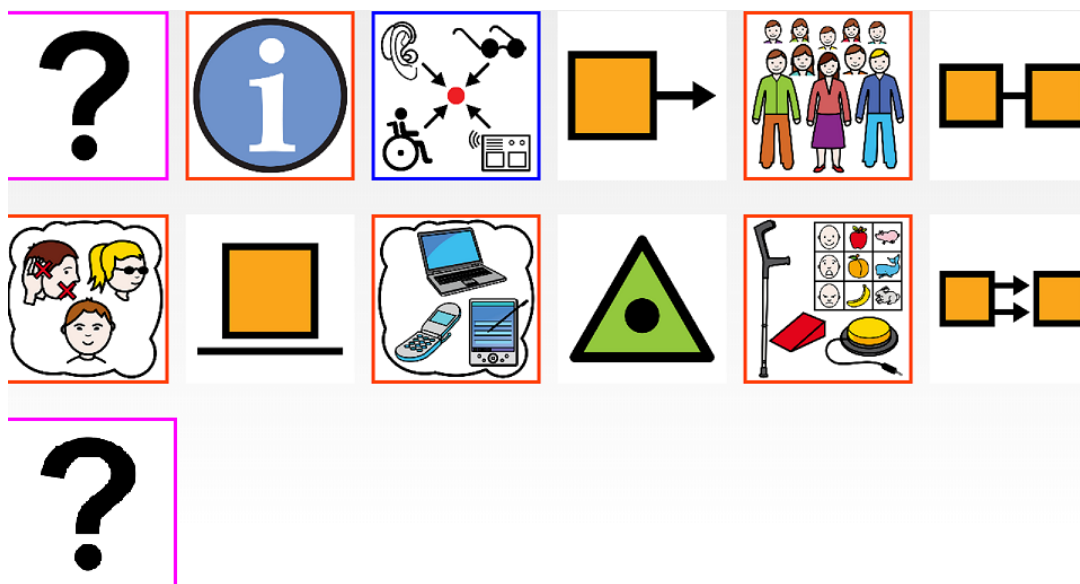
Declaración de los Derechos Humanos:



PictoTraductor:



AraTraductor:

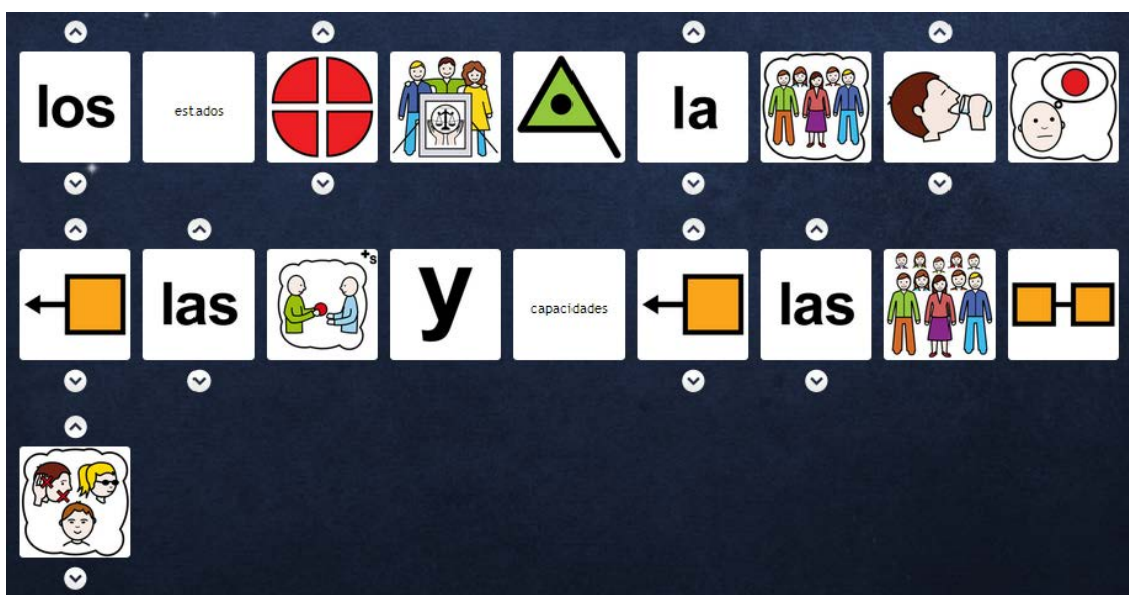


FRASE 5: "Los estados partes promoverán que la sociedad tome conciencia de las aportaciones y las capacidades de las personas con discapacidad."

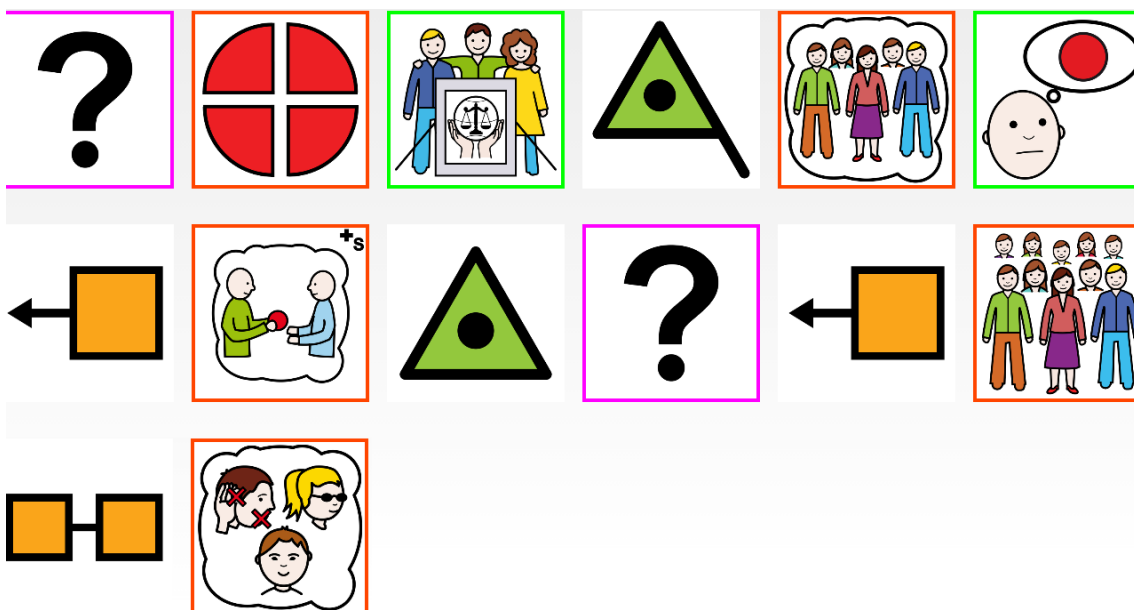
Declaración de los Derechos Humanos:



PictoTraductor:

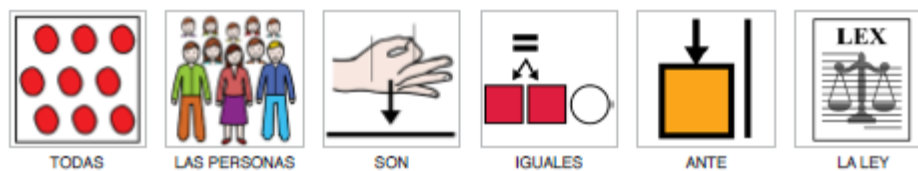


AraTraductor:

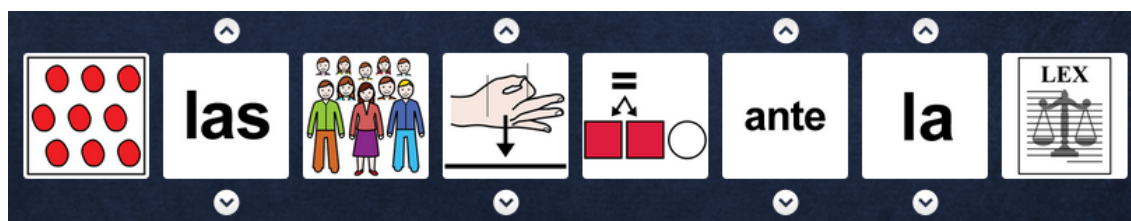


FRASE 6: "Todas las personas son iguales ante la ley."

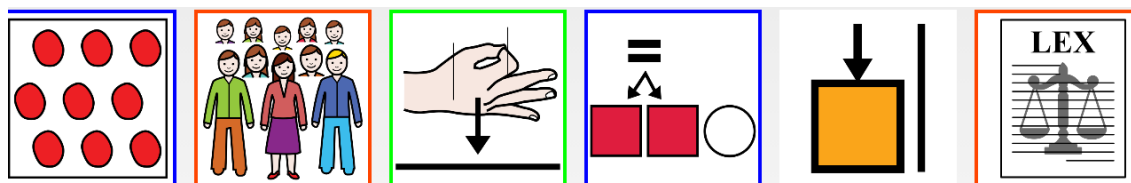
Declaración de los Derechos Humanos:



PictoTraductor:

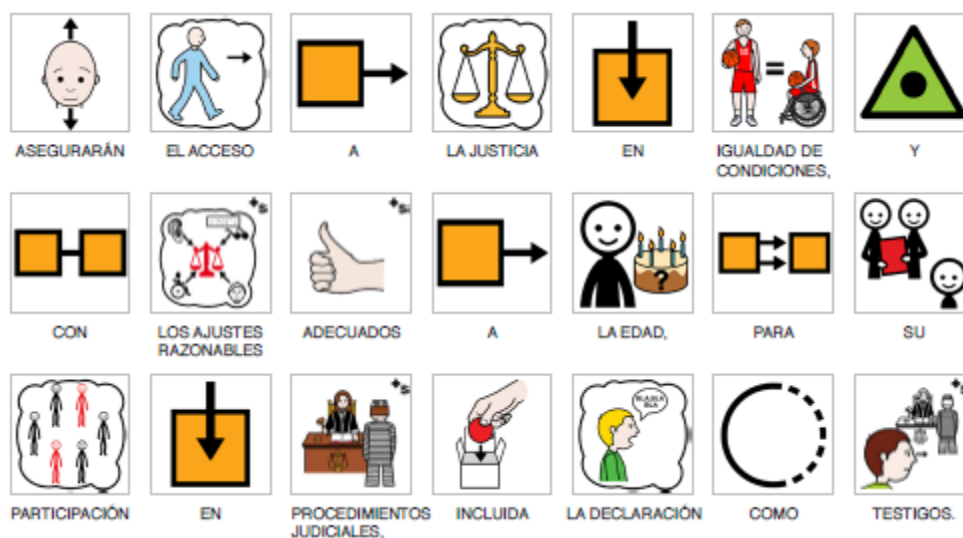


AraTraductor:



FRASE 7: "Asegurarán el acceso a la justicia en igualdad de condiciones y con los ajustes razonables adecuados a la edad, para su participación en procedimientos judiciales incluida la declaración como testigos."

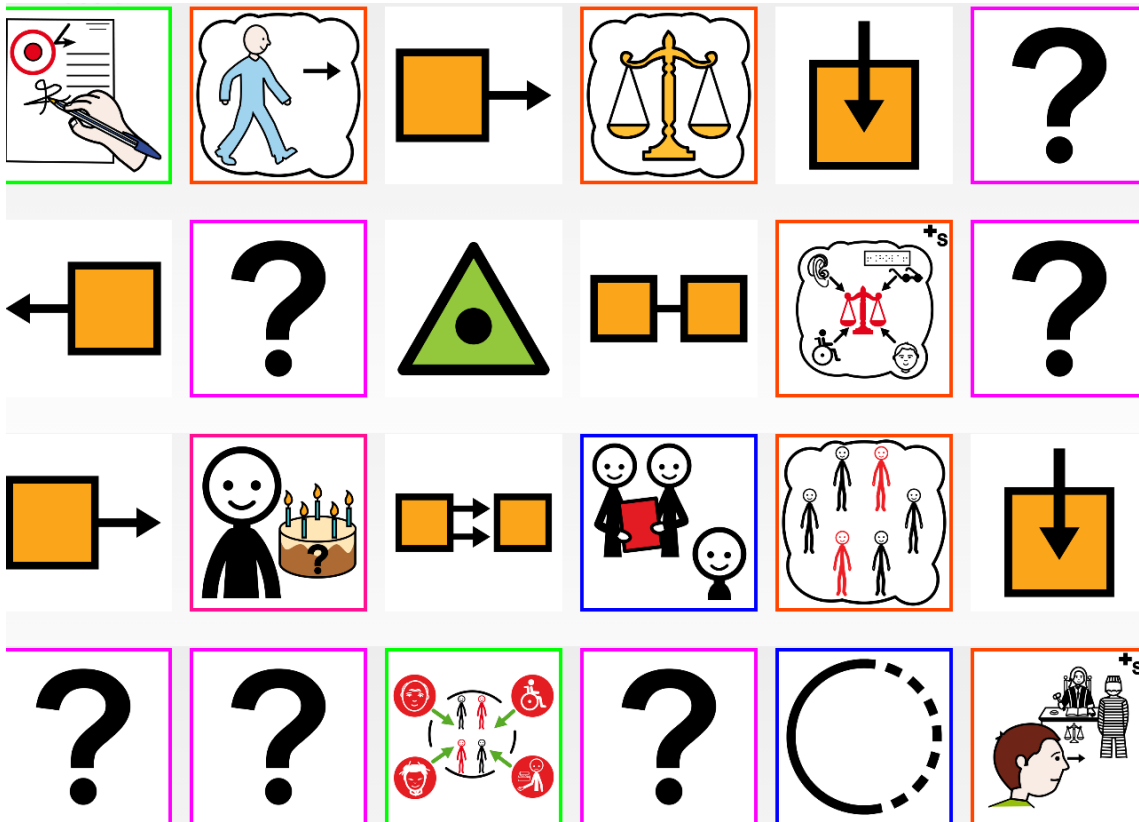
Declaración de los Derechos Humanos:



PictoTraductor:



AraTraductor:

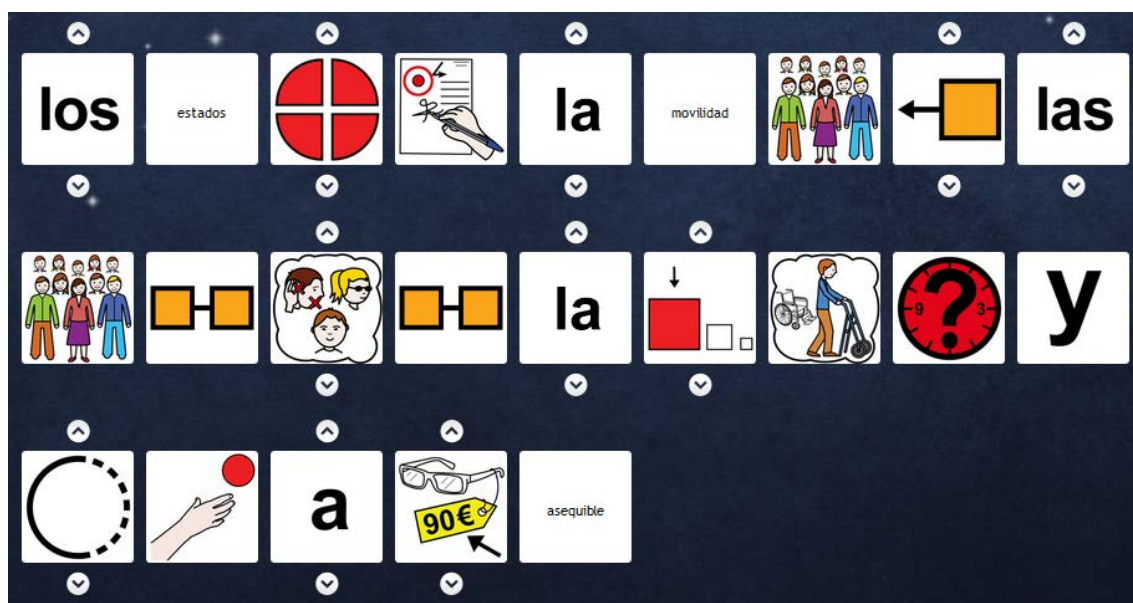


FRASE 8: "Los estados partes asegurarán la movilidad personal de las personas con discapacidad con la mayor independencia cuando y como deseen a precio asequible."

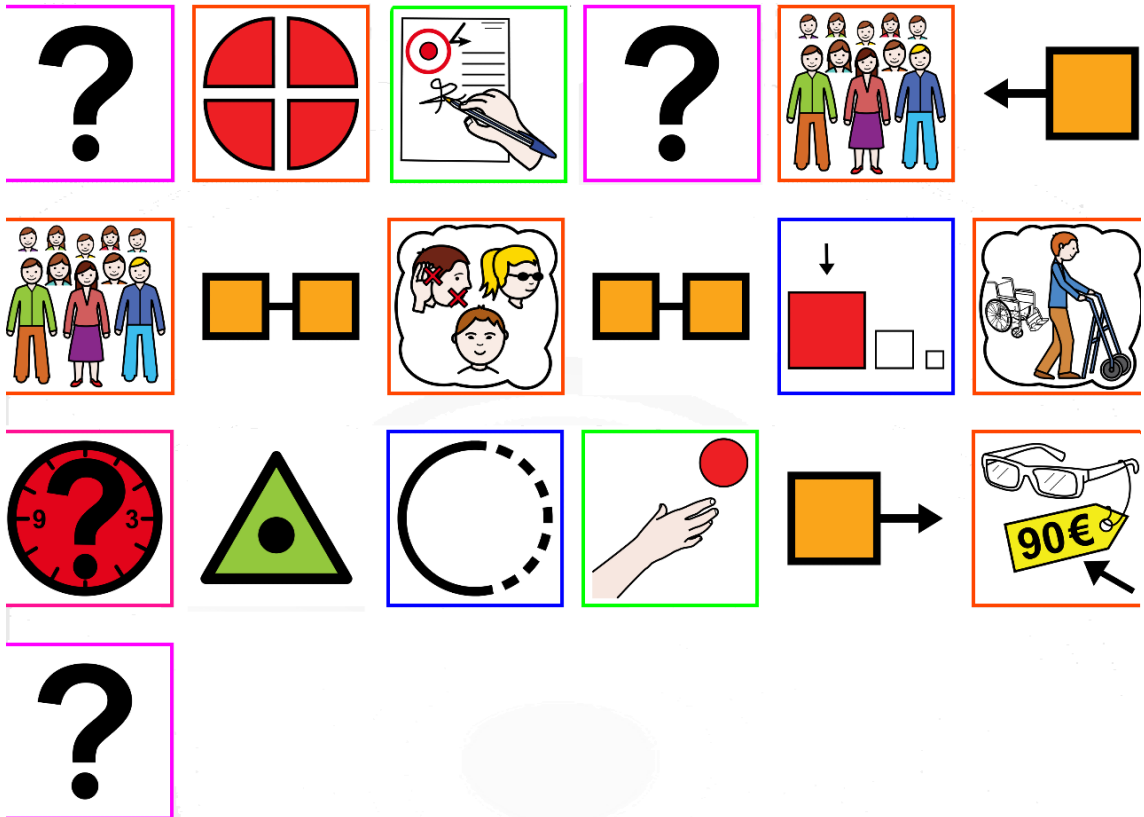
Declaración de los Derechos Humanos:



PictoTraductor:



AraTraductor:



FRASE 9: "Los estados partes vigilarán que se hagan los ajustes razonables para las personas con discapacidad en el lugar de trabajo."

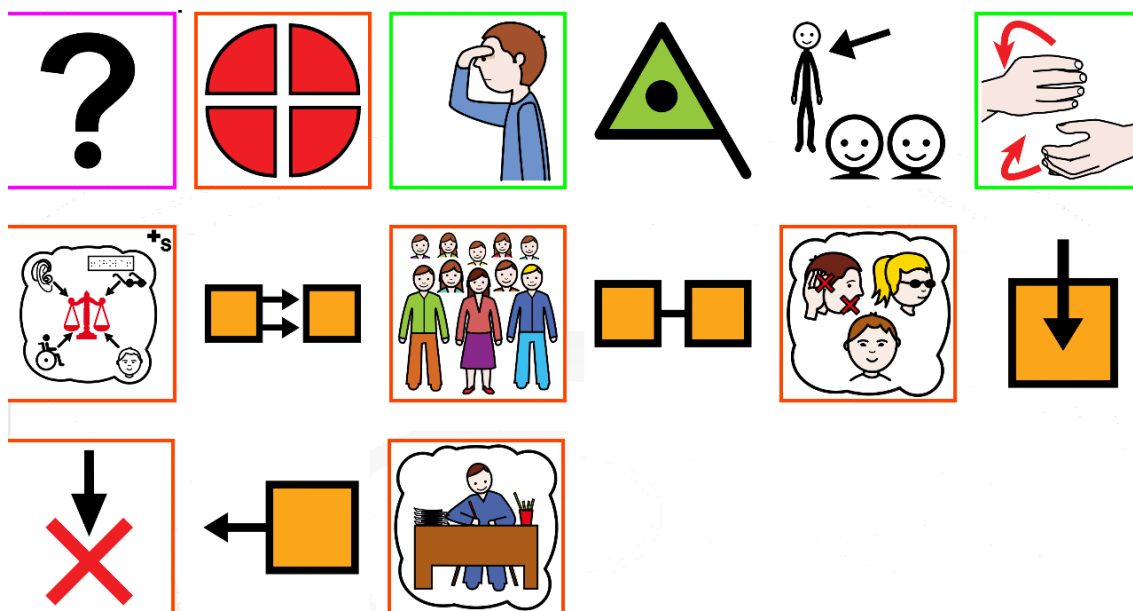
Declaración de los Derechos Humanos:

| | | | | |
|--------------------|-------------------------------|-----|----------|------------------------|
| | | | | |
| LOS ESTADOS PARTES | VIGILARÁN | QUE | SE HAGAN | LOS AJUSTES RAZONABLES |
| | | | | |
| PARA | LAS PERSONAS CON DISCAPACIDAD | EN | EL LUGAR | DE |
| | | | | |
| TRABAJO; | | | | |

PictoTraductor:

| | | | | | | | | |
|------------|---------|-----|--|--|----|--|-----|--|
| los | estados | | | | se | | los | |
| razonables | | las | | | | | el | |
| | | | | | | | | |

AraTraductor:

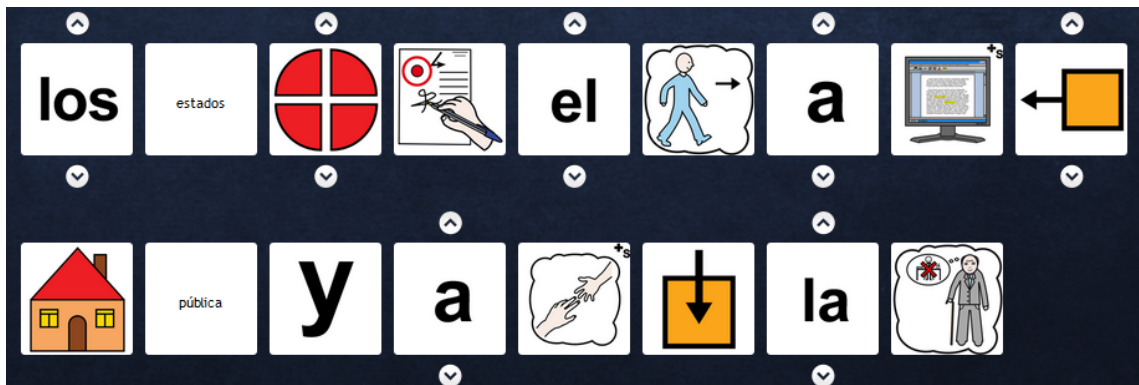


FRASE 10: "Los estados partes asegurarán el acceso a programas de vivienda pública y a apoyos en la jubilación."

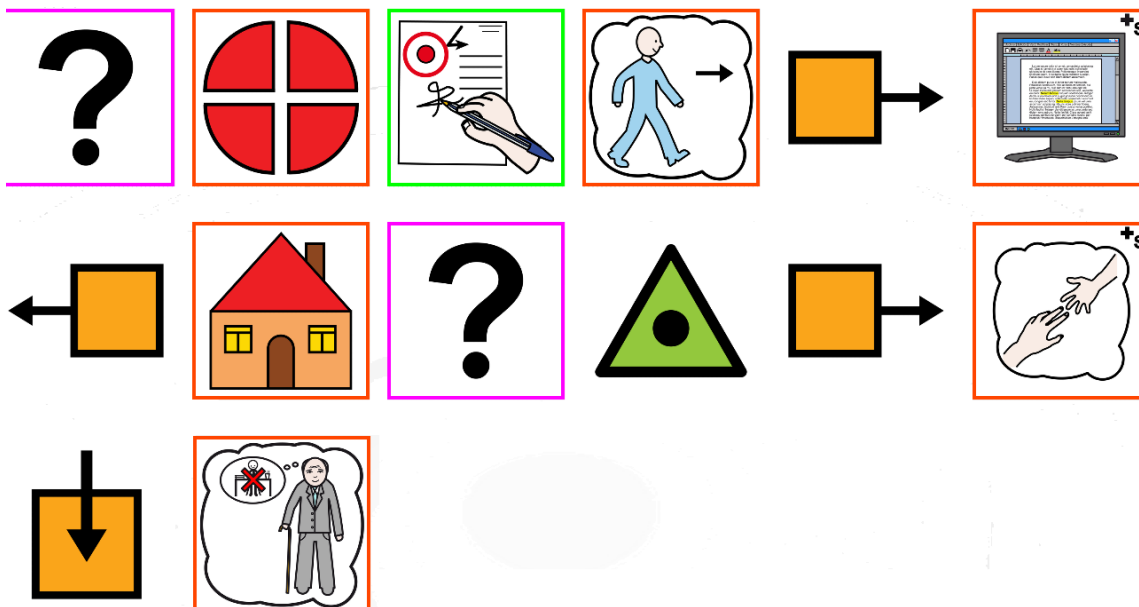
Declaración de los Derechos Humanos:



PictoTraductor:

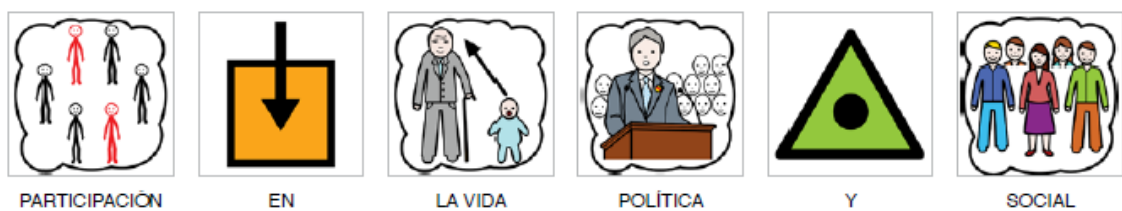


AraTraductor:



FRASE 11: "Participación en la vida política y social."

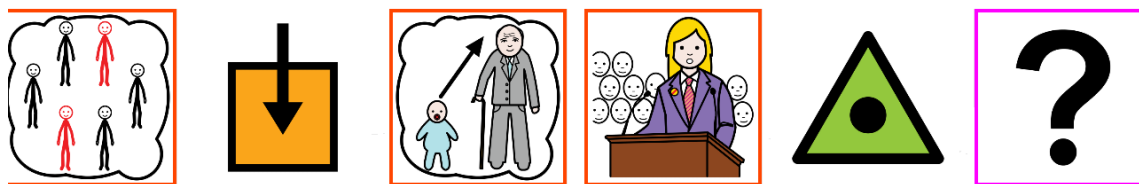
Declaración de los Derechos Humanos:



PictoTraductor:



AraTraductor:



FRASE 12: "Las personas con discapacidad tienen el derecho a que se proteja la información sobre su vida privada, familiar, hogar, correspondencia, salud y rehabilitación en igualdad de condiciones."

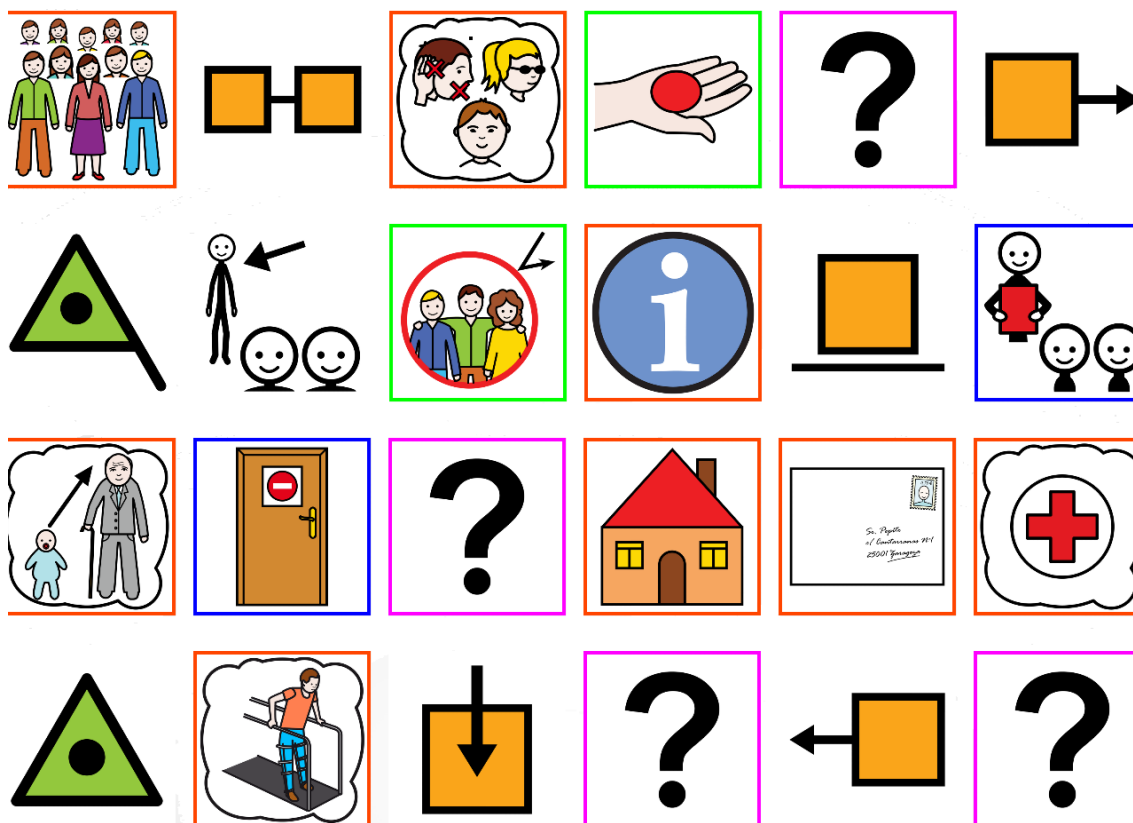
Declaración de los Derechos Humanos:



PictoTraductor:

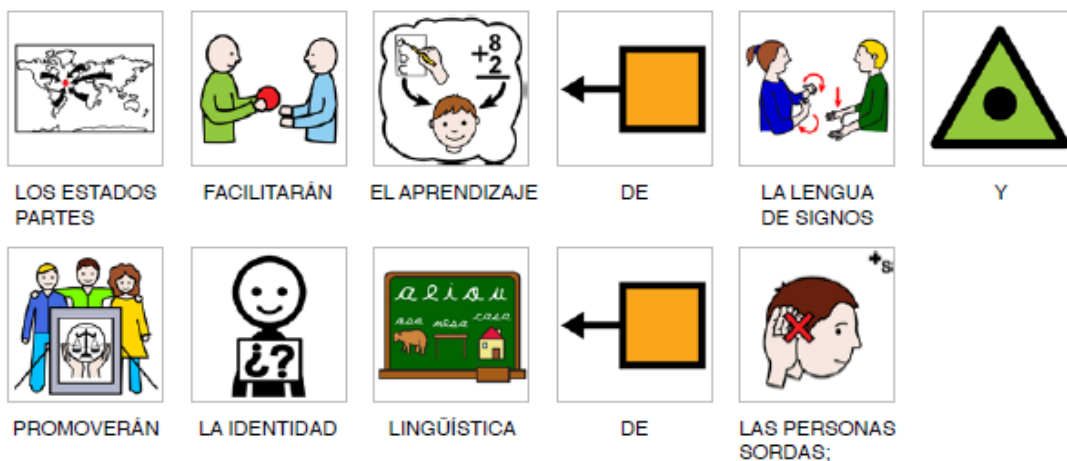


AraTraductor:



FRASE 13: "Los estados partes facilitaran el aprendizaje de la lengua de signos y promoverán la identidad lingüística de las personas sordas"

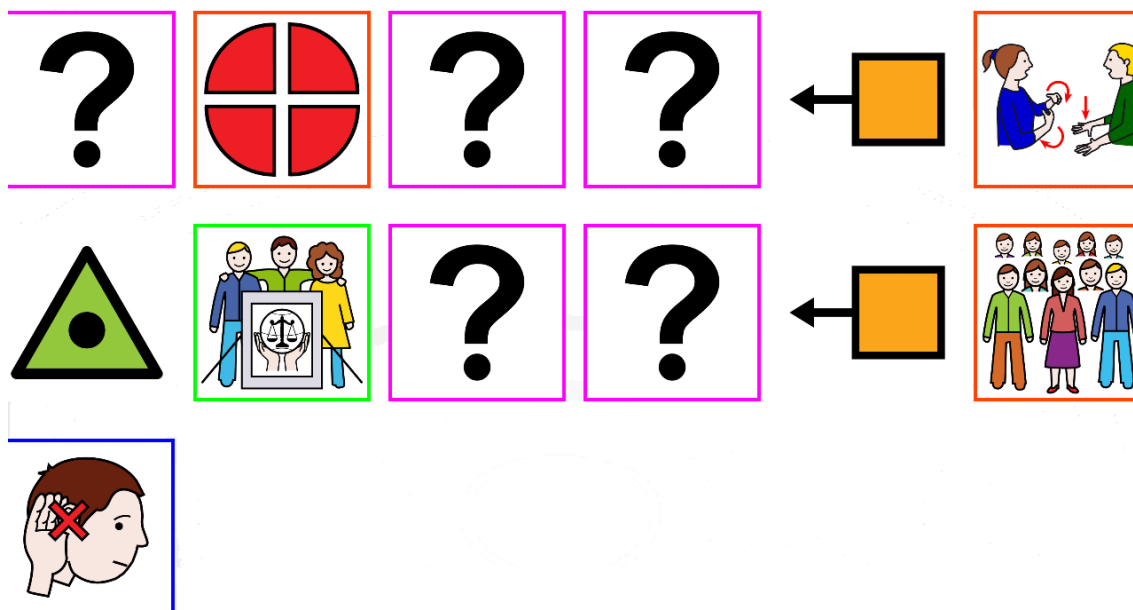
Declaración de los Derechos Humanos:



PictoTraductor:



AraTraductor:

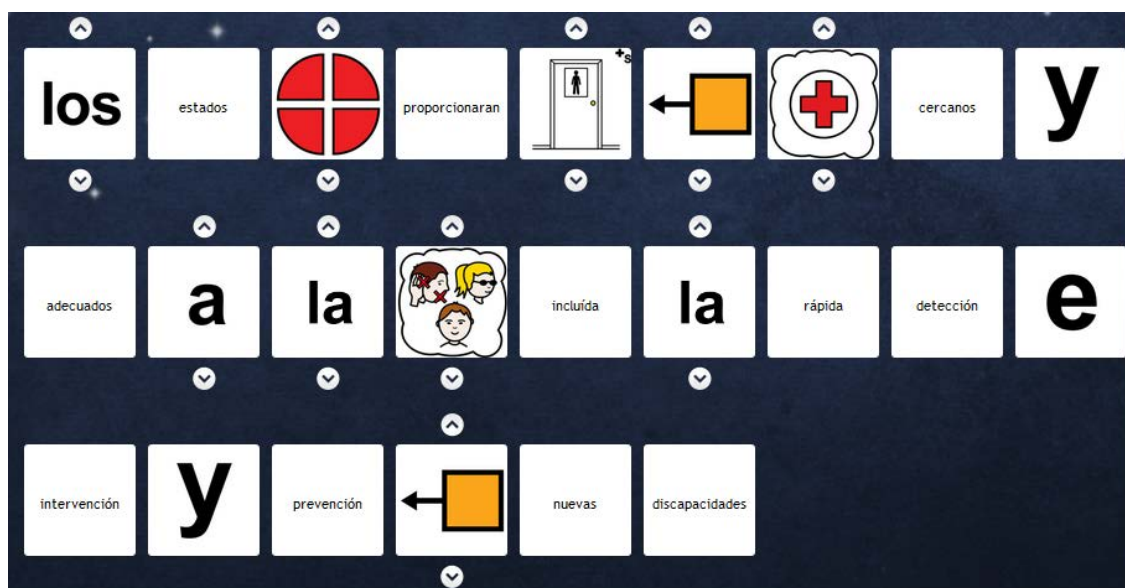


FRASE 14: "Los estados partes proporcionaran servicios de salud cercanos y adecuados a la discapacidad incluida la rápida detección e intervención y prevención de nuevas discapacidades."

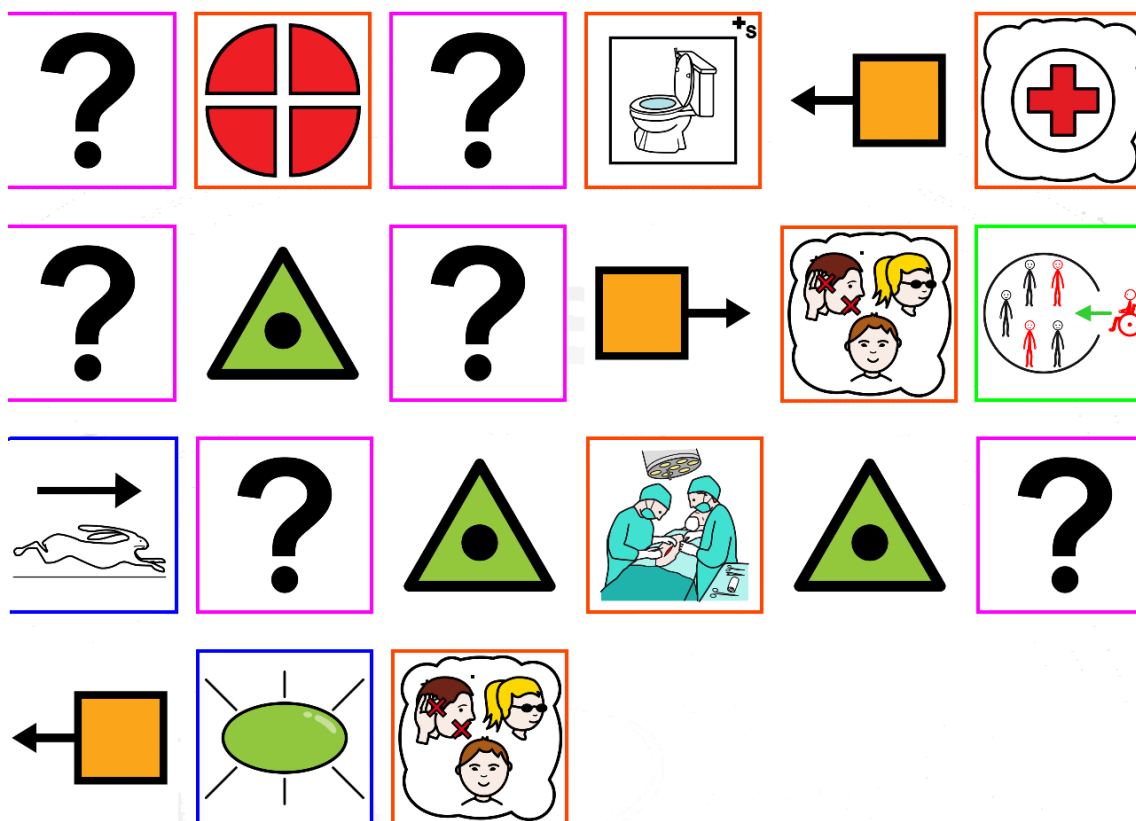
Declaración de los Derechos Humanos:



PictoTraductor:

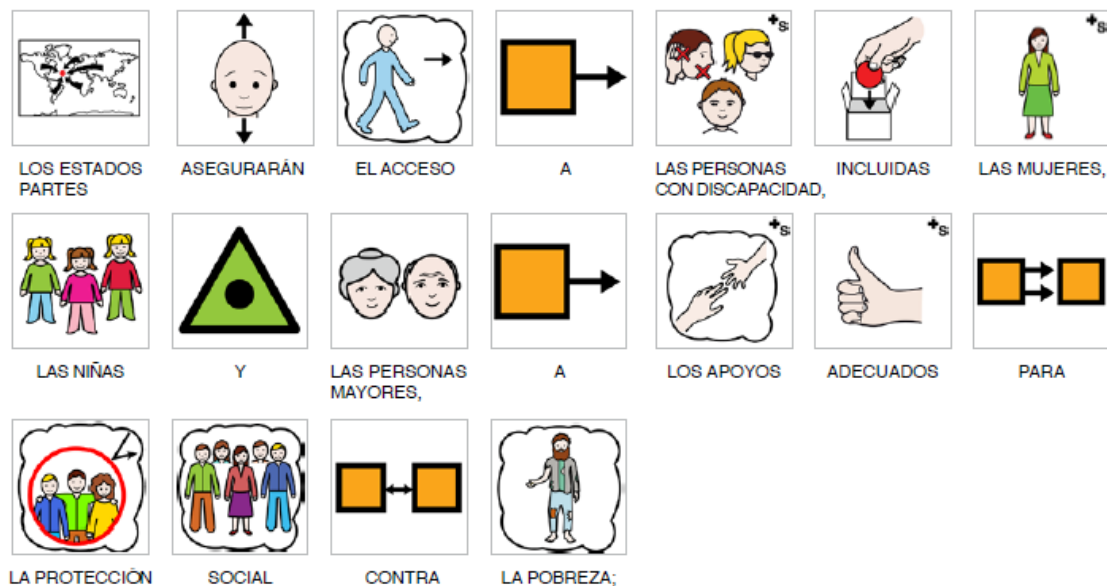


AraTraductor:



FRASE 15: "Los estados partes aseguran el acceso a las personas con discapacidad incluidas las mujeres, las niñas y las personas mayores a los apoyos adecuados para la protección social contra la pobreza."

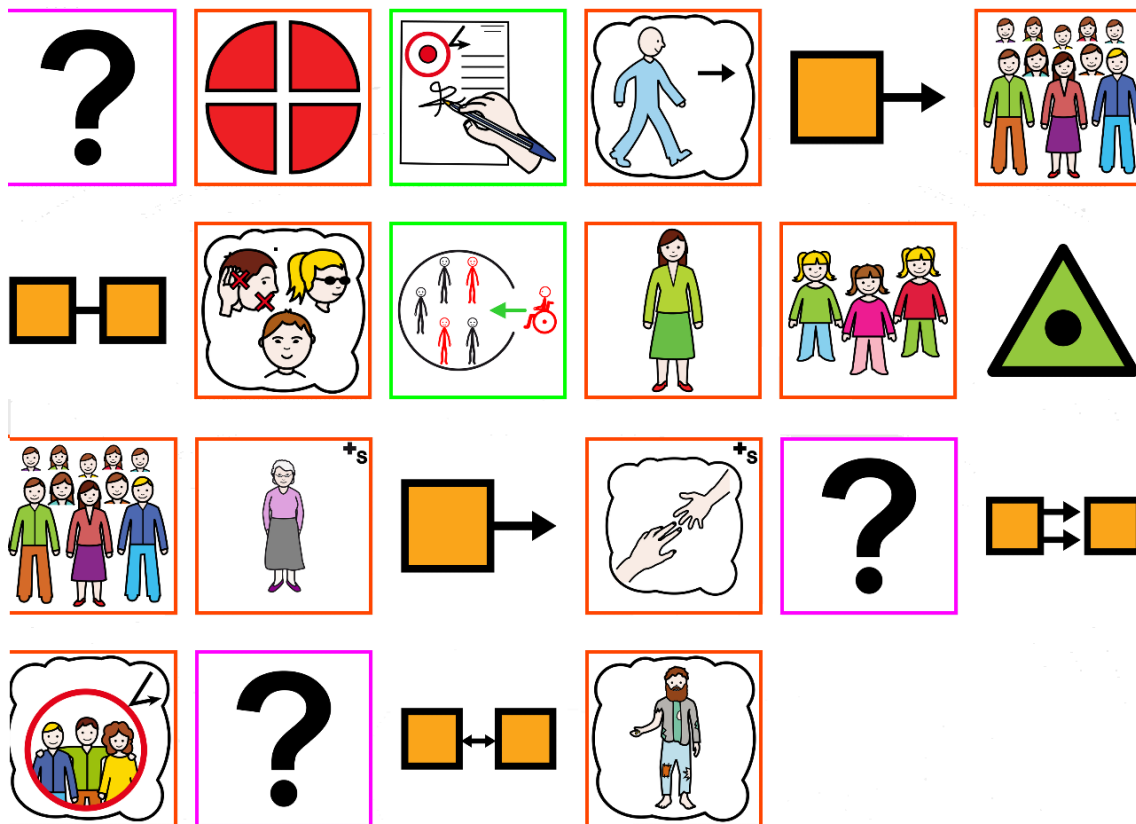
Declaración de los Derechos Humanos:



PictoTraductor:



AraTraductor:



FRASE 16: "Participación en actividades culturales, de tiempo libre y deporte."

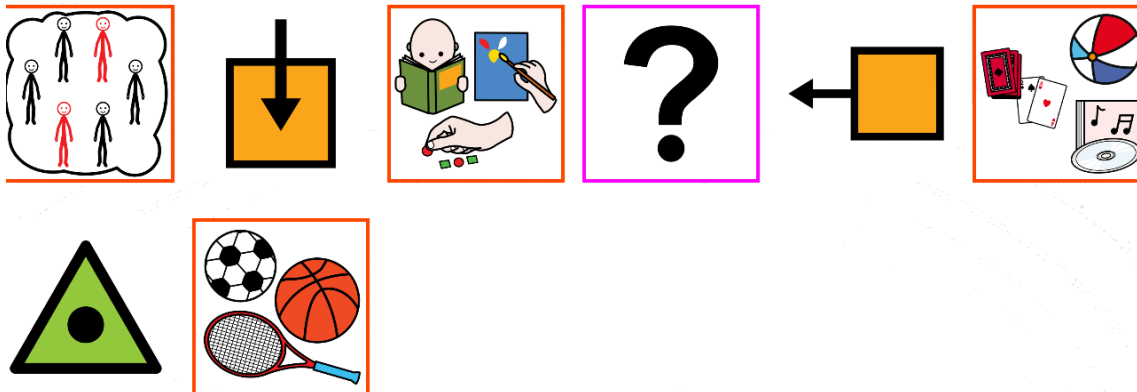
Declaración de los Derechos Humanos:



PictoTraductor:

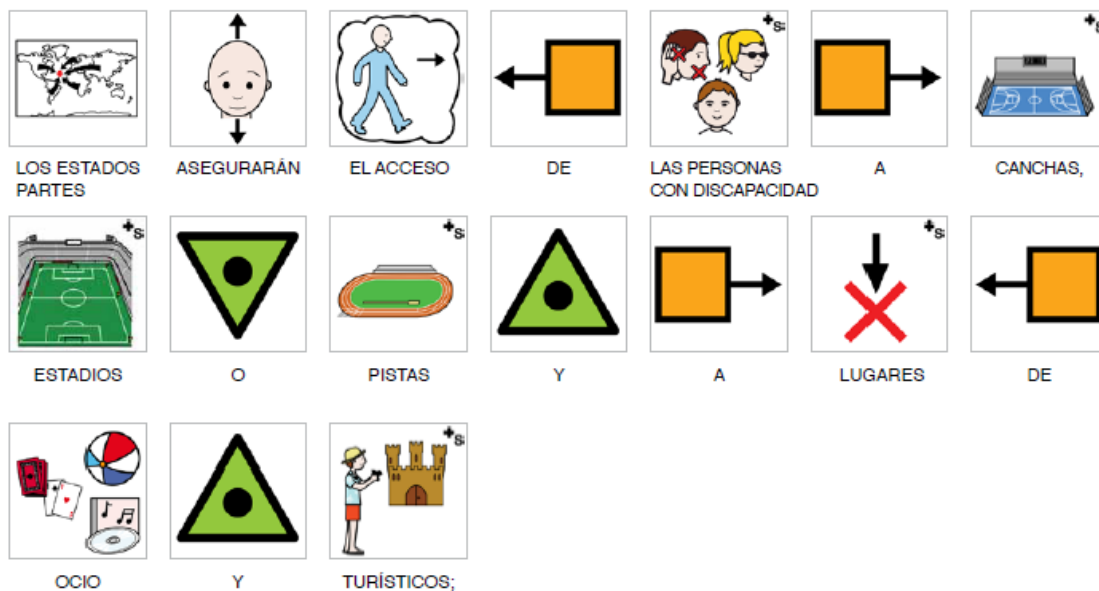


AraTraductor:



FRASE 17: "Los estados partes aseguran el acceso de las personas con discapacidad a canchas, estadios o pistas y a lugares de ocio y turísticos."

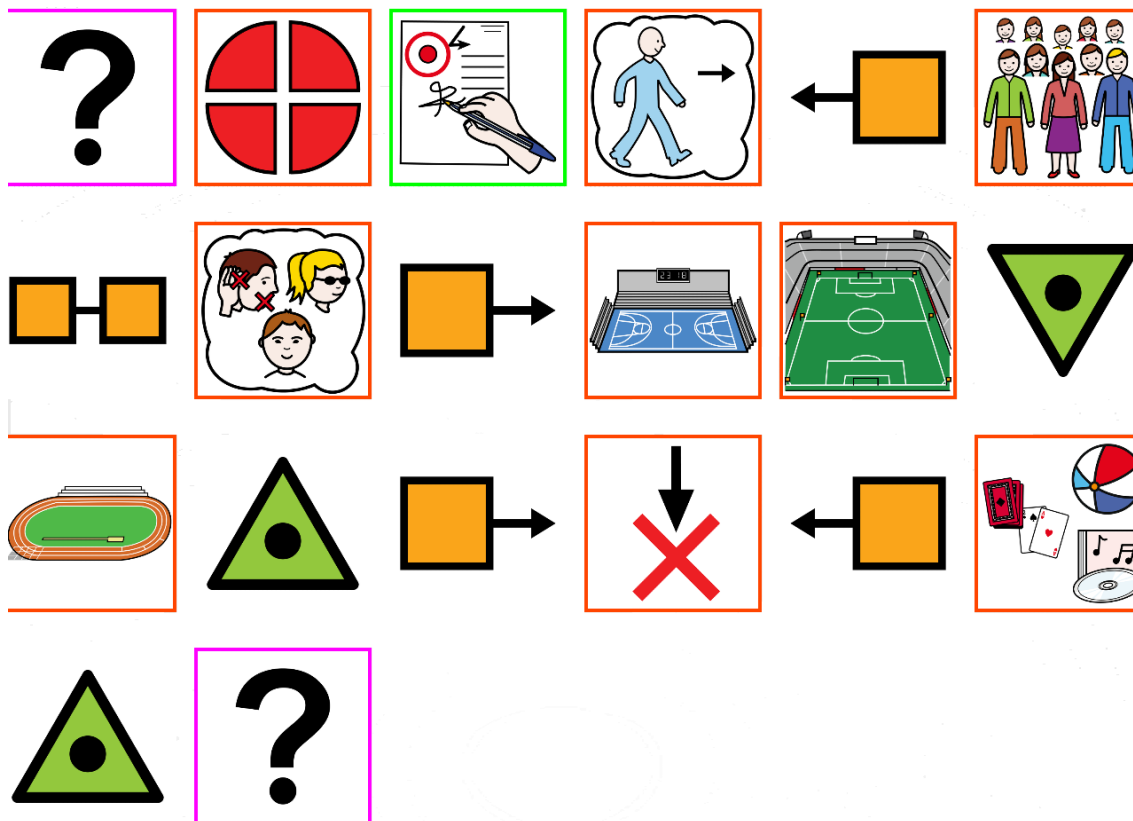
Declaración de los Derechos Humanos:



PictoTraductor:

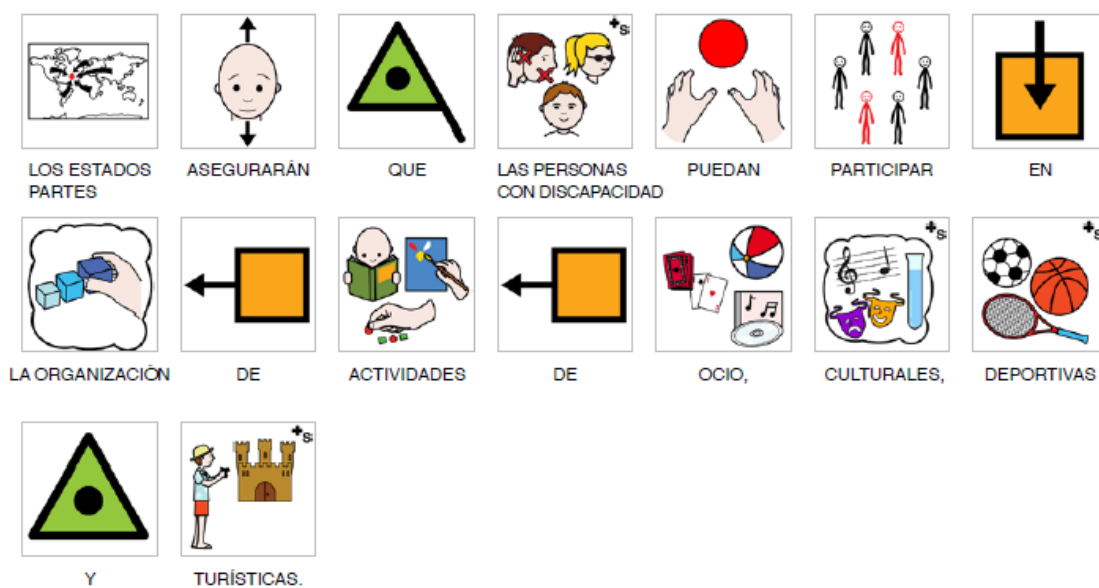


AraTraductor:

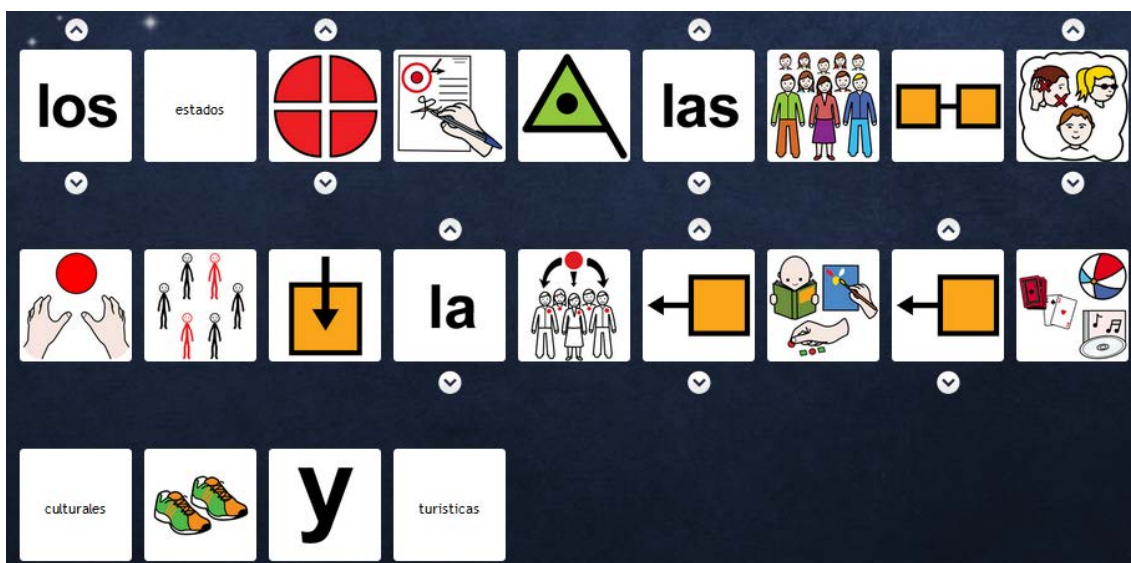


FRASE 18: "Los estados partes aseguran que las personas con discapacidad puedan participar en la organización de las actividades de ocio, culturales, deportivas y turísticas."

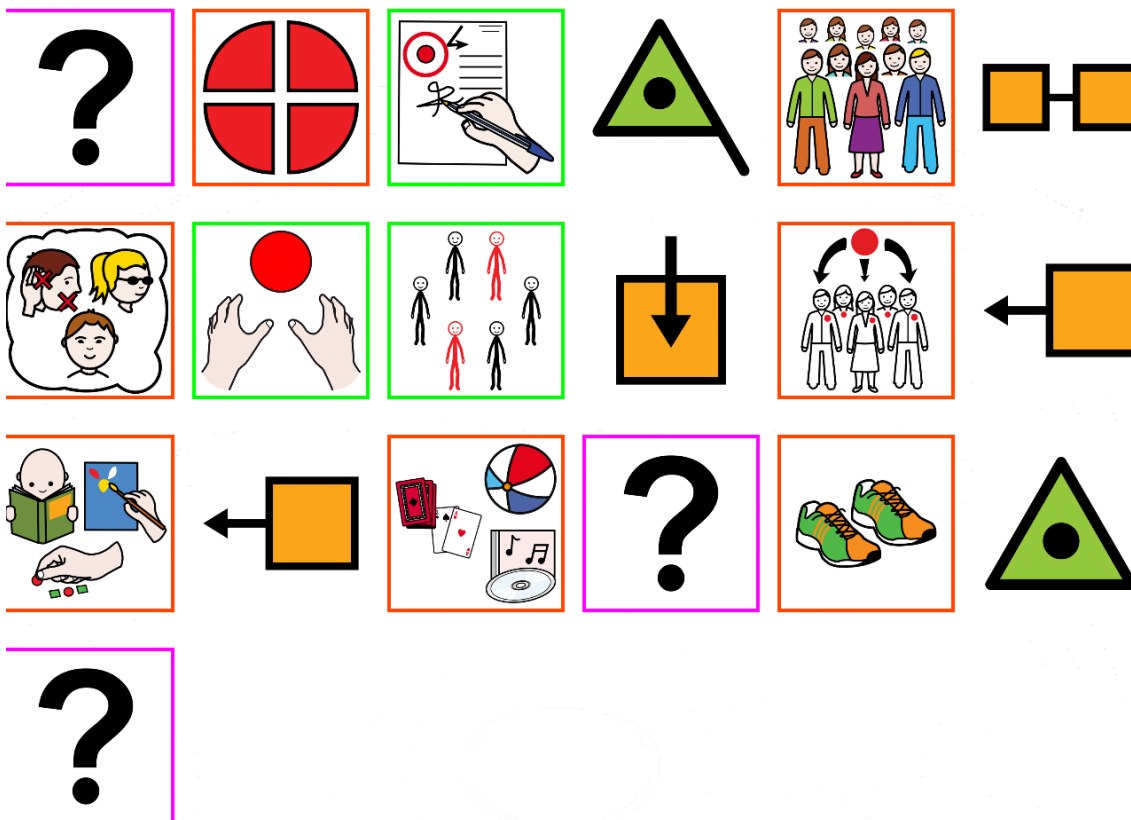
Declaración de los Derechos Humanos:



PictoTraductor:



AraTraductor:






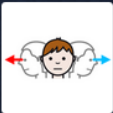







Frase 19: "Tú recuerda preguntar a las personas de tu entorno las dudas o los pictogramas que no entiendas."

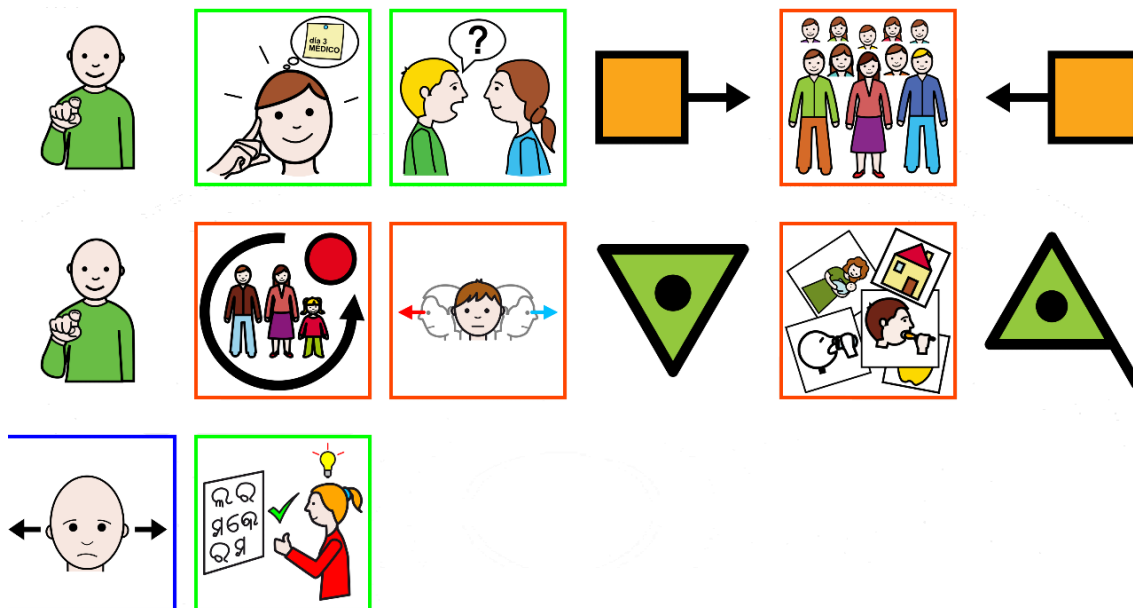
Declaración de los Derechos Humanos:

| | | | | |
|--|--|--|--|---|
|  |  |  |  |  |
| TÚ | RECUERDA | PREGUNTAR | A | LAS PERSONAS |
|  |  |  |  |  |
| DE | TU | ENTORNO | LAS DUDAS | O |
|  |  |  | | |
| LOS PICTOGRAMAS | QUE | NO ENTIENDAS | | |

PictoTraductor:

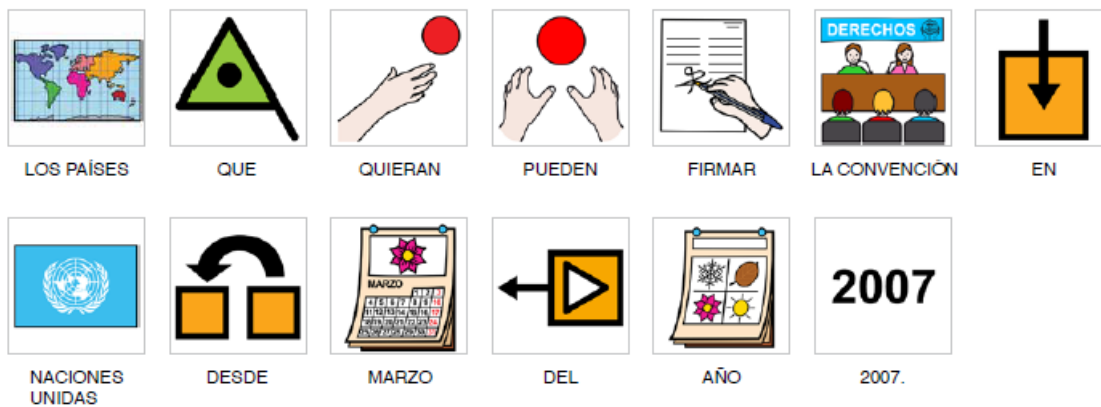
| | | | | | | | | |
|-----|---|---|-----|---|--|---|---|---|
| tu |  |  | a | las |  |  | TU |  |
| las |  |  | los |  |  |  |  | |

AraTraductor:

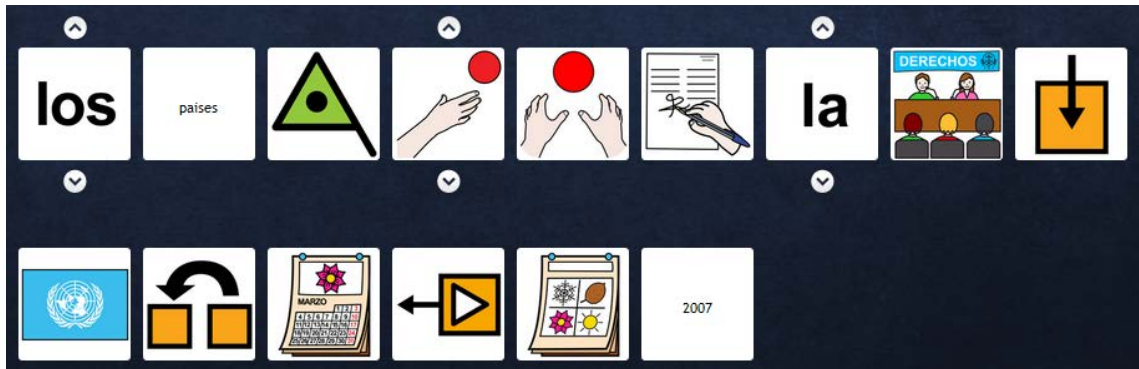


Frase 20: "Los países que quieran pueden firmar la convención en Naciones Unidas desde Marzo del año 2007."

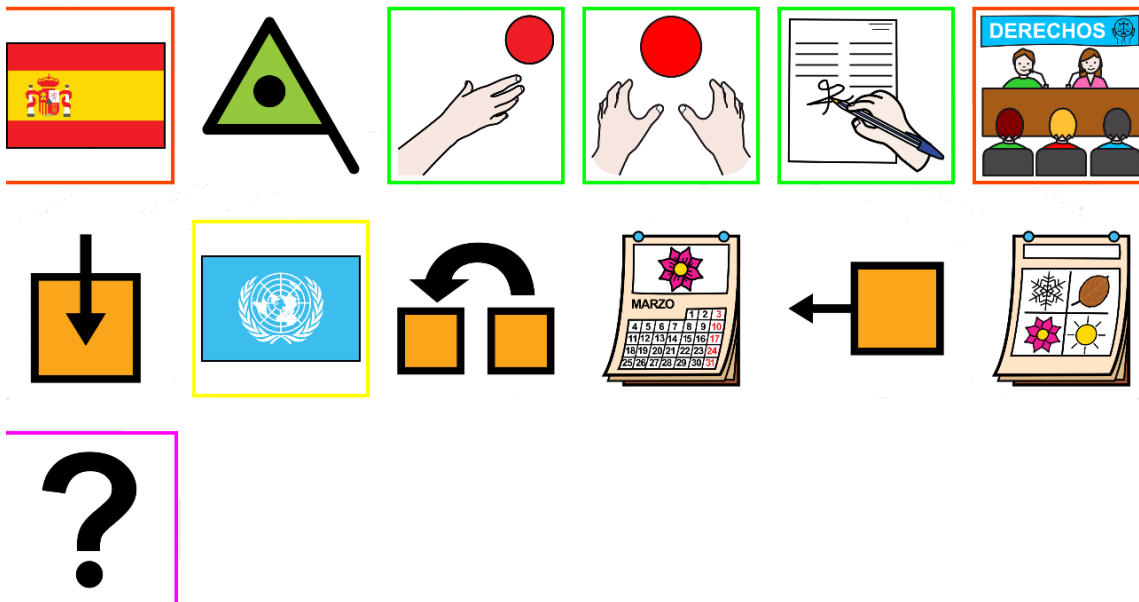
Declaración de los Derechos Humanos:



PictoTraductor:

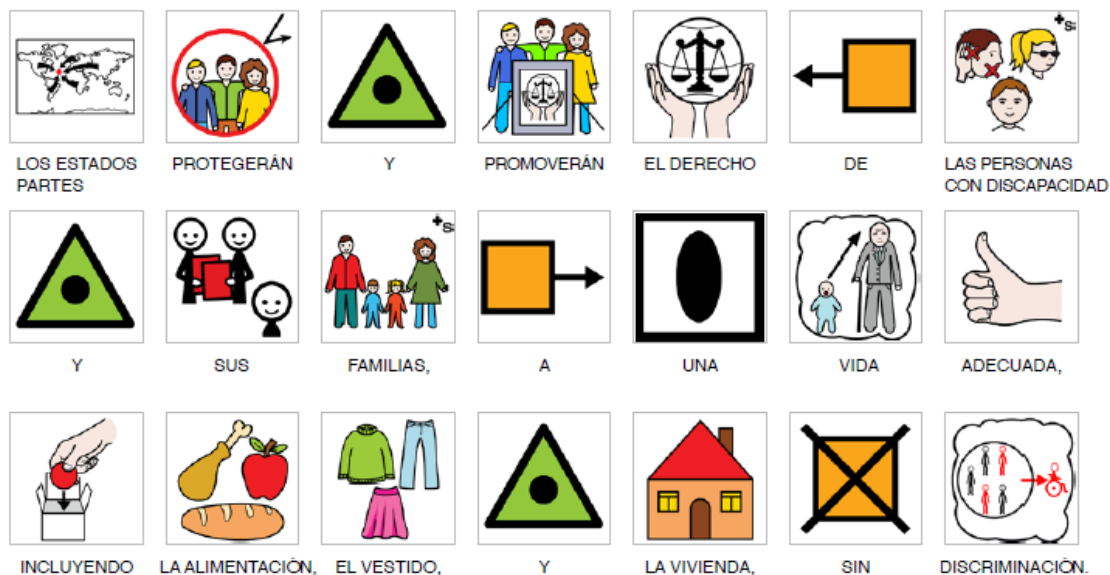


AraTraductor:



FRASE 21: "Los estados partes protegerán y promoverán el derecho de las personas con discapacidad y sus familias a una vida adecuada incluyendo la alimentación, el vestido, y la vivienda sin discriminación."

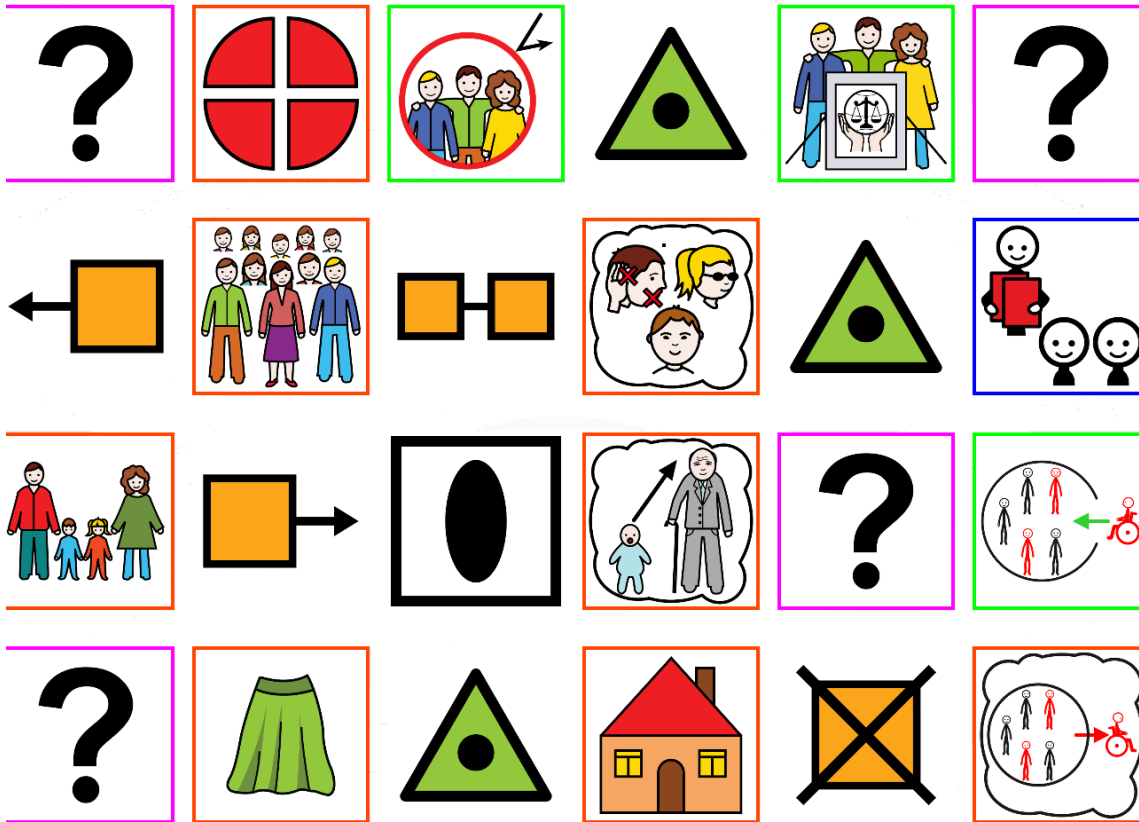
Declaración de los Derechos Humanos:



PictoTraductor:



AraTraductor:





Capítulo 11: Bibliografía



11. Bibliografía

Android. *Página web de Android*. <http://www.android.com/> (último acceso: 14 de 06 de 2014).

Apache Software Foundation. *Página web de Apache Tomcat*. <http://tomcat.apache.org/> (último acceso: 14 de 06 de 2014).

—. *Página web de Axis2*. <http://axis.apache.org/axis2/java/core/> (último acceso: 14 de 06 de 2014).

AraBoard. *Página web de AraBoard*. <http://giga.cps.unizar.es/affectivelab/araboard.html> (último acceso: 14 de 06 de 2014).

ARASAAC. *ARASAAC*. <http://www.catedu.es/arasaac/> (último acceso: 14 de 06 de 2014).

—. *Portal Aragonés de la Comunicación Aumentativa y Alternativa*. <http://www.arasaac.org/> (último acceso: 12 de 06 de 2014).

AraSuite. *Página web de AraSuite*. <http://sourceforge.net/projects/arasuite/?source=recommended> (último acceso: 14 de 06 de 2014).

AraWord. *Página web de AraWord*. <http://araword.sourceforge.net/> (último acceso: 14 de 06 de 2014).

Ballesteros, Miguel. «Mejora de la precisión para el análisis de dependencias usando Maltparser para el castellano.» Trabajo de Fin de Máster, UCM, 2010.

Johan Hall, Jens Nilsson and Joakim Nivre. *Maltparser*. <http://www.maltparser.org/>.

Ksoap2 Android. <https://code.google.com/p/ksoap2-android/wiki/HowToUse> (último acceso: 10 de 06 de 2014).

Oracle. *Desarrollo del patrón DAO por Oracle*. <http://www.oracle.com/technetwork/java/dataaccessobject-138824.html> (último acceso: 14 de 06 de 2014).

Pictoaplicaciones. *Pictotraductor*. <http://www.pictotraductor.com/> (último acceso: 14 de 06 de 2014).



Pivotal Software. *Página web de Spring*.
<http://docs.spring.io/spring/docs/2.0.x/reference/mvc.html> (último acceso: 14 de 06 de 2014).

Redhat. *Página oficial de Hibernate*. <http://hibernate.org/> (último acceso: 14 de 06 de 2014).

Ruiz Pacheco, Juan Carlos. *Inyección de Dependencias*.
<http://msdn.microsoft.com/es-es/library/jj635998.aspx> (último acceso: 14 de 06 de 2014).

Sandra Baldassarri, Javier Marco Rubio, Marta García Azpiroz, Eva Cerezo.
«AraBoard: A Multiplatform Alternative and Augmentative Communication Tool.»
Procedia Computer Science 27 (2013): 197-206.

W3C. *SOAP*. <http://www.w3.org/TR/soap12-part0/> (último acceso: 14 de 06 de 2014).

—. *WSDL*. <http://www.w3.org/TR/wsdl> (último acceso: 14 de 06 de 2014).